Máster en Ingeniería de Sistemas y Servicios Accesibles para la Sociedad de la Información

Trabajo Fin de Máster		
Título	Implementación de un entorno de mantenimiento centralizado para el diagnóstico y aislamiento de fallos en plataformas de sistemas modulares	
Autor	Roberto Sobrino Solís	
Tutor	Dr. Miguel Ángel Sánchez-Puebla Rodríguez	VºBº
Ponente	Dr. Eduardo Barrera López de Turiso	
Tribunal		
Presidente	Dr. Mariano Ruiz González	
Secretario	Dr. Francisco Javier Jiménez Martínez	
Vocal	Dr. Manuel Vázquez López	
Fecha de lectura	9 de Julio de 2015	
Calificación		

El Secretario:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

Máster en Ingeniería de Sistemas y Servicios Accesibles para la Sociedad de la Información

TRABAJO FIN DE MASTER

IMPLEMENTACIÓN DE UN ENTORNO DE MANTENIMIENTO CENTRALIZADO PARA EL DIAGNÓSTICO Y AISLAMIENTO DE FALLOS EN PLATAFORMAS DE SISTEMAS MODULARES

ROBERTO SOBRINO SOLÍS

JULIO 2015

A mis padres, que siempre me apoyan en mis proyectos,
a Mirian, que me ha soportado durante estos meses,
a Miguel Angel, que me ha animado a seguir estudiando,
y a los que me han mostrado su apoyo durante este tiempo.

Índice

Índice de figuras

Índice de tablas

Índice de ecuaciones

Resumen

En la actualidad gran parte de las industrias utilizan o desarrollan plataformas, las cuales integran un número cada vez más elevado de sistemas complejos. El mantenimiento centralizado permite optimizar el mantenimiento de estas plataformas, por medio de la integración de un sistema encargado de gestionar el mantenimiento de todos los sistemas de la plataforma.

Este Trabajo Fin de Máster (TFM) desarrolla el concepto de mantenimiento centralizado para sistemas complejos, aplicable a plataformas formadas por sistemas modulares. Está basado en la creciente demanda de las diferentes industrias en las que se utilizan este tipo de plataformas, como por ejemplo la industria aeronáutica, del ferrocarril y del automóvil.

Para ello este TFM analiza el Estado del Arte de los sistemas de mantenimiento centralizados en diferentes industrias, además desarrolla los diferentes tipos de arquitecturas de sistemas, las técnicas de mantenimiento aplicables, así como los sistemas y técnicas de mantenimiento basados en funciones de monitorización y auto diagnóstico denominadas *Built-In-Test Equipment* (BITE).

Adicionalmente, este TFM incluye el desarrollo e implementación de un modelo de un Entorno de Mantenimiento Centralizado en LabVIEW. Este entorno está formado por el modelo de un Sistema Patrón, así como el modelo del Sistema de Mantenimiento Centralizado y la interfaces entre ellos.

El modelo del Sistema de Mantenimiento Centralizado integra diferentes funciones para el diagnóstico y aislamiento de los fallos. Así mismo, incluye una función para el análisis estadístico de los datos de fallos almacenados por el propio sistema, con el objetivo de proporcionar capacidades de mantenimiento predictivo a los sistemas del entorno.

Para la implementación del modelo del Entorno de Mantenimiento Centralizado se han utilizado recursos de comunicaciones vía TCP/IP, modelización y almacenamiento de datos en ficheros XML y generación automática de informes en HTML.

Summary

Currently several industries are developing or are making use of multi system platforms. These platforms are composed by many complex systems. The centralized maintenance allows the maintenance optimization, integrating a maintenance management system. This system is in charge of managing the maintenance dialog with the different and multiple platforms.

This Master Final Project (TFM) develops the centralized maintenance concept for platforms integrated by modular and complex systems. This TFM is based on the demand of the industry that uses or develops multi system platforms, as aeronautic, railway, and automotive industries.

In this way, this TFM covers and analyzes several aspects of the centralized maintenance systems like the State of the Art, for several industries. Besides this work develops different systems architecture types, maintenance techniques, and techniques and systems based on Built-in-test Equipment functions.

Additionally, this TFM includes a LabVIEW Centralized System Environment model. This model is composed by a Standard System, the Centralized Maintenance System and the corresponding interfaces.

Several diagnostic and fault isolation functions are integrated on the Centralized Maintenance Systems, as well a statistic analysis function, that provides with predictive maintenance capacity, based on the failure data stored by the system.

Among others, the following resources have been used for the Centralized System Environment model development: TCP/IP communications, XML file data modelization and storing, and also automatic HTML reports generation.

1 Introducción

En la industria se puede definir mantenimiento como el conjunto de actividades y acciones que permiten que un equipo, útil o herramienta esté en condiciones de prestar servicio. Estas actividades incluyen reparaciones, modificaciones, inspecciones y acciones encaminadas a determinar si el dispositivo bajo mantenimiento funciona de acuerdo a su especificación [1].

A parte de restaurar la operación de un sistema, otros objetivos de las actividades de mantenimiento son tanto proporcionar ayuda para la detección y aislamiento de fallos como verificar la correcta operación del sistema, evitando de este modo paradas durante la operación, y sus costes asociados, reduciéndose también el número de repuestos y reparaciones necesarios.

El motor principal en la evolución de las técnicas y métodos de mantenimiento a lo largo de la historia ha sido aumentar el tiempo de operación de un sistema, disminuyendo los costes de mantenimiento [2], [3], lo cual se traduce en la reducción de costes en el ciclo de vida de un sistema.

Históricamente el mantenimiento de un sistema se realizaba empleando el método de inspección visual. Aunque éste fue el primer método de mantenimiento empleado, hoy en día se sigue siendo utilizado para el mantenimiento de gran número de sistemas porque no requiere de un diseño adicional en el mismo, aunque sus resultados son muy limitados.

Con el paso del tiempo, con el mayor conocimiento de los sistemas y basándose en el conocimiento adquirido, se han desarrollado métodos de mantenimiento programado. Con estos métodos de mantenimiento se determinan los intervalos en los cuales se deben realizar las tareas de mantenimiento a un determinado equipo.

Si bien hay diferentes teorías (curva de la bañera, mortalidad infantil,…) [2] que definen los modos de fallo en los equipos en función del tiempo, desde hace algunos años, y en contra de estas teorías, se están comprobando de forma empírica, que el mantenimiento programado no es todo lo efectivo que se podría esperar. Aunque estadísticamente la mayor parte de los equipos fallan en torno a un intervalo de tiempo determinado, existe un porcentaje de equipos que durarán más de la media y al igual que otro porcentaje que durarán menos tiempo de la media.

La utilización de un método de mantenimiento programado, como por ejemplo el método MSG3 [4], implica que al finalizar el tiempo de operación determinado para un equipo, pese a que dicho equipo no haya presentado ninguna condición o síntoma de fallo, se procede a ejecutar la tarea de mantenimiento aplicable a dicho equipo. Esta tarea de mantenimiento puede ser una tarea de calibración, reparación, sustitución, etc. Si esta tarea de mantenimiento no se hubiera aplicado según indica el mantenimiento programado, es probable que el equipo en cuestión hubiera seguido funcionando en condiciones óptimas un tiempo adicional hasta llegar a presentar alguna condición de fallo.

Partiendo de este hecho se están desarrollando métodos de mantenimiento predictivo [4]. Estos métodos de mantenimiento se basan principalmente en la observación del funcionamiento de un equipo con el fin de detectar la variación no controlada de sus parámetros característicos de funcionamiento.

Teniendo en cuenta que la rápida evolución tecnológica promueve el incremento de la complejidad de los sistemas y la cada vez mayor integración entre ellos hacen que la detección de fallos se cada vez una tarea más compleja. Por este motivo, para implementar el concepto de mantenimiento predictivo es necesario recurrir a diferentes técnicas como procesado de señales, reconocimiento de patrones, comparación con datos empíricos, comparación con datos de modelos físicos o incluso redes neuronales.

La implementación del mantenimiento predictivo en un sistema, implica que cada sistema sea capaz de monitorizarse a sí mismo, comprobando que todas sus funciones y las diferentes partes o equipos del sistema funcionan adecuadamente. Esta capacidad se denomina *Built-In-Test Equipment* (BITE). Además, cada sistema dispone de la capacidad de aviso a los operadores del sistema o a otros equipos, en caso de que alguna función o parte del equipo presente alguna discrepancia en su funcionamiento.

Aunque inicialmente la idea de incluir capacidades BITE en un sistema pudiera parecer innecesaria, costosa y que aumenta la complejidad del sistema respecto a las estrategias de diagnóstico convencionales, con el paso del tiempo, se ha demostrado que es necesario implementar estas técnicas de mantenimiento en los sistemas. Algunos estudios indican que el tiempo invertido

en el desarrollo de sistemas de diagnóstico multiplica el tiempo ahorrado por el usuario final de un sistema [5], [6].

En la actualidad uno de los requisitos para el diseño y fabricación de cualquier sistema automatizado, derivado de la reducción de costes, es incluir en el sistema capacidades de auto diagnóstico y aislamiento de fallos para aumentar la calidad de servicio de sistema y reducir su mantenimiento y la dependencia de la intervención técnicos humanos.

Aprovechando las capacidades de auto diagnóstico de los sistemas junto a las elevadas capacidades de integración entre sistemas, tenemos el entorno necesario para desarrollar un Sistema de Mantenimiento Centralizado. El Sistema de Mantenimiento Centralizado permite la gestión de las funciones de mantenimiento de los distintos equipos conectados a través de interfaces dedicadas a este sistema [7], [8].

Hoy en día, los sistemas de mantenimiento centralizados son una solución aceptada por diferentes industrias, cuyo objetivo principal es mejorar la eficacia del mantenimiento de una plataforma multisistema. Las principales funciones de un Sistema de Mantenimiento Centralizado deben estar orientadas al diagnóstico y aislamiento de fallos para los múltiples sistemas que forman la plataforma. Las capacidades de diagnóstico y aislamiento de fallos están basadas en las estrategias BITE [7].

Con el fin de desarrollar los entornos de mantenimiento centralizados, este TFM se ha dividido en las siguientes secciones:

- Objetivos:

  Este TFM presenta e implementa una aproximación para el mantenimiento centralizado de sistemas complejos, a través de una solución basada en la tecnología BITE. También propone los patrones que en los que deberían basarse los equipos y el protocolo de comunicación de los equipos. La implementación desarrollada en este TFM se basa en estos patrones propuestos.

- Estado del Arte:

  Esta sección analiza a los actuales y principales métodos y técnicas de mantenimiento de sistemas complejos embarcados en

plataformas, tanto en el campo de la aeronáutica, como en el campo ferroviario y automovilístico.

- Técnicas de Mantenimiento de Sistemas Complejos:

  Tras analizar el estado del arte, en esta sección se desarrollan los principales tipos de arquitecturas de diseño de sistemas, las principales técnicas de mantenimiento, y las técnicas de BITE aplicables a los diferentes tipos de sistemas.

- Entorno de Mantenimiento Centralizado:

  A continuación, en esta sección se presenta el mantenimiento de sistemas complejos basado en el concepto de Mantenimiento Centralizado. Para ello se desarrolla la arquitectura del entorno y los diferentes sistemas que lo componen, al igual que las funciones del sistema de mantenimiento incluyendo una función de procesado estadístico de los datos de los fallos almacenados, basada en la distribución de probabilidad de Weibull.

- Implementación en LabVIEW de un modelo de un Entorno de Mantenimiento Centralizado:

  En esta sección se documenta el modelo desarrollado del Entorno de Mantenimiento Centralizado, analizando su arquitectura, las funciones implementadas en los sistemas, las interfaces, las técnicas y los recursos utilizados, al igual que la propia operación del entorno. Para la implementación de este entorno se ha utilizado LabVIEW como herramienta de modelización ampliamente utilizada y convertida en un estándar de facto.

- Evaluación:

  En esta sección se analiza el cumplimiento de los objetivos presentados en este TFM, e incluye la evaluación del correcto funcionamiento del modelo desarrollado, tanto de los patrones de los sistemas propuestos, de las funciones implementadas así como del protocolo de comunicación y mensajes intercambiados entre sistemas.

- Conclusiones:

  En último lugar de este TFM, se recogen y se presentan las conclusiones sobre los resultados obtenidos, tanto en la facilidad para la implementación basada en los patrones de sistemas propuestos y su protocolo, así como la capacidad del modelo implementado para detectar, aislar y reportar fallos. En esta sección también se incluye la posibilidad de aplicación de este TFM a las industrias aeronáutica, ferroviaria y automovilística. También se incluyen los trabajos futuros que permiten ampliar el concepto de Sistema Patrón presentado en este TFM.

2 Objetivos

Este TFM está basado en una de las principales necesidades de la industria actualmente y en el futuro próximo. Las estimaciones indican que en un plazo de entre los próximos 20 - 30 años la mayor parte de los procesos de mantenimiento de los equipamientos industriales dispondrán de un entorno de mantenimiento centralizado.

El objetivo principal de este TFM es presentar e implementar una aproximación para el mantenimiento centralizado de sistemas complejos en plataformas multisistemas. Este concepto está basado en las funciones de diagnóstico y aislamiento de fallos de los sistemas, las cuales se desarrollan a partir de la tecnología *Built-In-Test Equipment* (BITE).

Para conseguir dicho objetivo este TFM desarrolla una arquitectura patrón para un entorno de mantenimiento centralizado, junto a una serie de funciones que facilitan el diagnóstico y aislamiento de fallos de los diferentes sistemas embarcados en la plataforma, incluyendo una función orientada al mantenimiento predictivo, basada en el procesado estadístico de la distribución de Weibull. Adicionalmente proporciona un modelo en LabVIEW de este entorno de mantenimiento centralizado.

Conjuntamente a este objetivo principal, este TFM incluye como objetivos adicionales:

Proporcionar una visión integral de la evolución de la arquitectura de sistemas utilizados en la industria, en relación con los métodos y técnicas de mantenimiento, junto con el desarrollo de las principales técnicas BITE.

Desarrollar las principales técnicas de BITE implementadas en los diferentes sistemas de las plataformas industriales, tanto en aplicaciones software como en aplicaciones hardware.

Presentar los beneficios de la implementación de una funcionalidad BITE en un sistema complejo usado en plataformas industriales, y su impacto en el ciclo de vida de un sistema complejo.

3 Estado del arte

En industrias en las cuales se utilizan plataformas complejas con múltiples sistemas embarcados, desde hace algunos años, muchos de estos sistemas disponen funciones de monitorización y detección de fallos basados en estrategias BITE. De este modo, a través de estas funciones BITE, cada sistema de forma individual, es capaz de indicar si está operativo, o si por el contrario presenta algún fallo y está operando en modo degradado, o no está operativo.

Desde el punto de vista de mantenimiento de estas plataformas la implantación de estas tecnologías suponía un avance respecto los métodos de mantenimiento tradicionales, permitiendo la monitorización individual de los sistemas o equipos de una plataforma. No obstante dada la creciente complejidad y el aumento del número de sistemas en una plataforma, el mantenimiento aún sigue siendo bastante costoso.

Hoy en día, las últimas generaciones de sistemas en plataformas complejas, integran sistemas centralizados de mantenimiento. La base principal del sistema de centralizado de mantenimiento reside en que todos los sistemas de la plataforma, los cuales envían su estado al sistema de mantenimiento. De este modo, desde un único sistema, se monitoriza y controla el estado de todos los sistemas de la plataforma.

Los sistemas de centralizados de mantenimiento además de la monitorización de fallos de los sistemas de la plataforma, incluyen otras funciones que ayudan al mantenimiento como la posibilidad de comandar test a cada sistema de la plataforma, gestionar y almacenar los fallos que produce cada sistema a lo largo del tiempo, etc.

La clave para la implementación de un Sistema de Mantenimiento Centralizado en una plataforma en la que están integrados sistemas de múltiples fabricantes está en que todas las funciones e interfaces de mantenimiento de cada sistema deben utilizar la misma filosofía de BITE y los mismos protocolos de comunicación. En consecuencia, las funciones e interfaces de mantenimiento de los sistemas de una plataforma deben estar estandarizadas.

Para ello las diferentes industrias que utilizan este tipo de plataformas, están desarrollando normas con el fin de que los fabricantes de sistemas, desarrollen sistemas que incorporen estrategias de mantenimiento de BITE compatibles e integrables con un Sistema de Mantenimiento Centralizado (SMC).

A continuación se describe el estado del arte de las principales industrias en las cuales se utilizan Entornos de Mantenimiento Centralizados: aeronáutica, ferroviaria y automovilística.

3.2. Industria aeronáutica

Los fabricantes de sistemas para la industria aeronáutica desde los años 90 han desarrollado sistemas que incorporan funciones de detección de fallos. Inicialmente estas funciones de aislamiento de fallos eran particulares de cada sistema. Poco después, algunos fabricantes desarrollaron sistemas de mantenimiento que permitían la gestión del mantenimiento de varios sistemas de un mismo fabricante de forma automatizada y centralizada. Sin embargo esta solución omitía el mantenimiento del resto de sistemas de otros fabricantes integrados en una aeronave.

Como ejemplo, la mayoría de los sistemas de aviónica, con independencia del fabricante del avión (Airbus, Boeing, Embraer,…) disponen de una función de mantenimiento centralizado, la cual no es compatible con otros sistemas del avión de otros fabricantes y/o otras tecnologías.

Para conseguir que todos los fabricantes desarrollaran sus sistemas con unas funciones de monitorización de fallos estándar que permitieran la monitorización y control del mantenimiento estos sistemas desde un único sistema de mantenimiento, la organización ARINC ha desarrollado las normas ARINC 604 [7], y ARINC 624 [8].

3.2.1. ARINC 604 - *Guidance for design and use of Built-In Test Equipment*

La norma ARINC 604, recoge una serie de definiciones y recomendaciones relacionadas con el mantenimiento, el BITE y los sistemas de mantenimiento centralizados.

Con el fin de que los diferentes fabricantes de sistemas tengan un mismo criterio a la hora de plantear el mantenimiento de sus sistemas, el primer punto

de esta norma es definir los conceptos básicos del mantenimiento en sistemas de aeronaves. Entre estos conceptos, cabe destacar las siguientes ideas:

- En un avión, el nivel de básico de mantenimiento es a nivel equipo. Si un equipo está fallado, se reemplaza por otro, y el equipo fallado se envía a reparar al centro reparador correspondiente.

- Durante la operación de un avión, los sistemas están monitorizando su funcionamiento continuamente y en caso de que un sistema reporte algún fallo, este queda registrado para su posterior análisis durante el mantenimiento. En caso de que el sistema se degrade o deje de estar operativo se enviarán los avisos o alertas correspondientes a la operación del sistema, hasta que se pueda realizar la tarea de mantenimiento correspondiente.

3.2.1.1. Sistema centralizado de monitorización de fallos

La norma ARINC 604 introduce y define el concepto de sistema centralizado de monitorización de fallos desarrollando los siguientes puntos.

3.2.1.1.1. Arquitectura

El sistema centralizado de monitorización de fallos estará compuesto por los siguientes componentes:

- Unidad de procesado e interfaz con las particiones BITE de cada sistemas.

- Unidad de control y de indicación.

- Particiones BITE de cada sistema.

Además puede tener otros equipos opcionales como impresora, unidad de control e indicación portátil.

3.2.1.1.2. Funciones y aplicaciones

La norma ARINC 604 define las siguientes funciones y modos de operación para el centralizado de monitorización de fallos:

- Almacenamiento de los fallos:

  Es la función encargada de almacenar los fallos reportados por cada sistema de la plataforma, con el fin de gestionar el correcto mantenimiento de todos los sistemas.

- Equipos instalados: es la función encargada de gestionar un listado de los sistemas conectados a él, permitiendo el acceso a esta información por parte del operario de mantenimiento.

- Fallos reportados: es la función encargada de recopilar y almacenar los fallos reportados por la partición BITE de cada sistema, permitiendo el acceso a esta información durante las tareas de mantenimiento.

- Correlación de fallos:

  Es la función encargada de registrar tanto los fallos que han llevado asociado algún efecto en la operación de la aeronave, como el efecto en operación que ha producido ese fallo.

- Fallos intermitentes:

  El sistema centralizado de monitorización de fallos debe gestionar aquellos fallos que han sido recopilados por un sistema, aunque hayan sido reportados intermitentemente.

- Test de inicio:

  Es la función encargada de gestionar el resultado de los test de inicio de cada sistema, cada vez que un sistema se inicializa. Este test puede ser lanzado desde el sistema centralizado de monitorización de fallos.

- Comprobación de instalación:

  Esta función permite comprobar que después de reemplazar un equipo, éste está correctamente instalado.

- Test del sistema:

  Este test realiza una comprobación completa del sistema para el cual se ejecuta, reportando el código de fallo respectivo en caso de encontrar algún fallo en el sistema.

3.2.1.1.3. Interfaces

El protocolo de comunicación que define esta norma para las interfaces entre el sistema centralizado de monitorización de fallos y las particiones BITE de los diferentes sistemas del avión se basa en un bus ARINC 429 [9].

Además la norma también introduce una interfaz del sistema centralizado de monitorización de fallos con el sistema ACARS [9], con el fin de enviar los datos de mantenimiento al exterior del avión vía radio.

3.2.2. ARINC 624 - *Design guidance for Onboard Maintenance System*

La norma ARINC 624 define los conceptos y la arquitectura de un sistema de mantenimiento embarcado. Los principales puntos de esta norma son:

- Definición de los conceptos particulares relativos al mantenimiento.
- Definición de los objetivos que debe cumplir un sistema de mantenimiento embarcado.
- Definición de un Sistema de mantenimiento embarcado:
  - Arquitectura,
  - Funciones,
  - Interfaces.

3.2.2.1. Sistema de mantenimiento embarcado

La norma ARINC 624 introduce y define el concepto de sistema de mantenimiento embarcado, desarrollando los siguientes puntos.

3.2.2.1.1. Arquitectura

El sistema de mantenimiento embarcado estará compuesto por los siguientes equipos:

- Particiones BITE de cada sistema.
- Unidad de procesado central con interfaz con las particiones BITE de cada sistema.
- Terminal de acceso al mantenimiento.
- Interfaces con los siguientes sistemas:
  - sistemas de aviso en cabina.
  - documentación de mantenimiento.
  - Impresora.
  - enlace de datos.
  - Cargador/descargador de datos.
  - Selector de eventos.

Además puede incluir otros equipos opcionales como un terminal de mantenimiento remoto.

3.2.2.1.2. Funciones y aplicaciones

La norma ARINC 624 define las siguientes funciones y modos de operación para el sistema de mantenimiento embarcado:

- Test automáticos de test y aislamiento de fallos.

  El sistema de mantenimiento embarcado, tiene las siguientes funciones:

- Fallos reportados:

  Es la función encargada de monitorizar los sistemas conectados, recopilando y almacenado los fallos reportados por la partición BITE de cada sistema, permitiendo el acceso a esta información durante las tareas de mantenimiento.

- Correlación de fallos:

  Es la función encargada de registrar tanto los fallos que han llevado asociado algún efecto en la operación de la aeronave, como el efecto en operación que ha tenido ese fallo.

- Fallos intermitentes:

  El sistema de mantenimiento embarcado debe gestionar aquellos fallos que han sido recopilados por el sistema, pero actualmente no se están reportando.

En caso de que en el avión existan sistemas que no están conectados al sistema de mantenimiento embarcado, este tiene la capacidad de guiar a través de la Documentación de Mantenimiento, en la ejecución de test de forma manual.

- Test comandados por el operador.

  - Configuración de equipos instalados:

    El sistema gestiona los equipos a nivel de números de parte hardware y software, permitiendo el acceso a esta información por parte del operario de mantenimiento.

  - Test operacional:

    Es el encargado de comprobar que el sistema bajo test, tiene todos sus módulos operativos incluyendo las interfaces con el resto de sistemas.

- Test del sistema:

  Este test comprueba que el sistema se comporta de acuerdo a la especificación. Este tipo de test se suele utilizar tras el instalaciones iníciales o tras reparaciones importantes.

- Comprobación de instalación:

  Este test permite comprobar la correcta instalación de un equipo, después de su instalación.

- Aislamiento de fallos interactivo:

  Este test permite la ejecución de diferentes test en modo interactivo para la confirmación de fallos, o detección de fallos que no son detectados en modo automático.

- Alineamiento y ajustes:

  Esta función permite realizar el ajuste o alineamiento de los equipos que requieren estas tareas tras su sustitución.

Adicionalmente incluye como opción la posibilidad de monitorizar la información de las diferentes interfaces.

- Documentación de mantenimiento.

  El sistema de mantenimiento embarcado, debe incluir la documentación de mantenimiento en formato electrónico, con el fin de poder realizar la tarea de mantenimiento o despachar el avión de forma eficaz.

  La documentación disponible a través de esta función es:

  - Manuales de mantenimiento.
  - Catálogos de partes.
  - Diagramas de cableado.
  - Esquemas de los sistemas del avión.

- Guía para despachar el avión.
- Otra documentación adicional, necesaria para el mantenimiento.

- Monitorización de las condiciones del avión.
  - Adquisición de datos:

    El sistema de mantenimiento embarcado podrá monitorizar la información de los buses y señales entre sistemas bajo la demanda del operador en un momento concreto, y acceder a la información de fallos de los sistemas del avión.

  - Generación y gestión de informes:

    Esta función permite generar y gestionar los informes de mantenimiento con los distintos tipos de información disponible y su envío a través de los diferentes dispositivos de salida como la impresora, enlace de datos, dispositivo de almacenamiento de datos, etc.

    De igual forma ofrece la capacidad de programar la generación automática de informes de acuerdo a los parámetros definidos por el operador, incluso permite definir lógicas de control para la adquisición de dichos parámetros.

- Monitorización de eventos.

  Esta capacidad permite a los operadores comandar la monitorización instantánea de los parámetros del avión mediante un pulsador en cabina. De este modo si los operadores detectan algún comportamiento anormal de los sistemas, comandando esta función se obtiene el detalle del estado de los sistemas en ese momento.

  Para implementar esta función el sistema de mantenimiento embarcado activa una señal de Evento que llega a todos los

sistemas de la plataforma. En ese momento cada sistema debe almacenar el estado de sus parámetros característicos.

La información de eventos almacenada debe estar disponible para ser enviada a la impresora, dispositivo de almacenamiento de datos o a través del enlace de datos.

3.2.2.1.3. Interfaces

Esta norma establece que el protocolo de comunicaciones entre el sistema de mantenimiento embarcado pueda estar implementado sobre diferentes tipos de bus, como ARINC 429 [9], ARINC 629 [10], ARINC 636 [11], ARINC 646 [12], incluso otro tipo de buses siempre que se implemente el protocolo de comunicación de acuerdo a la norma.

3.3. Industria ferroviaria

Los fabricantes de sistemas para la industria ferroviaria al igual que los fabricantes de sistemas para otras industrias, desde hace años han desarrollado sistemas con funciones de detección de fallos incorporadas.

Pero es en la actualidad cuando los fabricantes de la industria del ferrocarril como Alstom, Talgo, Bombardier, están comenzado a desarrollar sistemas globales para la gestión del tren, los cuales incorporan sistemas de mantenimiento integrados. El principal sistema de este tipo es el *Train Control and Management System* (TCMS).

3.3.1. IEC 62580 – *On board Multimedia and Telematic Subsystem*

La *International Electrotechnical Commission* (IEC) desde hace algunos años está trabajando en el desarrollo de la norma IEC 62580 - *On board Multimedia and Telematic Subsystem*, con el objetivo de estandarizar el desarrollo de múltiples sistemas embarcados en los ferrocarriles. La norma IEC 62580, tiene un enfoque muy amplio que engloba múltiples sistemas. Esta norma está dividida en cinco partes.

IEC 62580, *On board Multimedia and Telematic Subsystem* (OMTS):

- IEC 62580-1 *General architecture* [13].
- IEC 62580-2, *Video surveillance*/CCTV.
- IEC 62580-3, *Driver and Crew orientated services*.
- IEC 62580-4, *Passenger orientated services*.
- IEC 62580-5, *Train Operator/Maintainer orientated services*.

Actualmente únicamente está aprobada la primera parte, IEC 62580-1, la cual establece de forma general la arquitectura de los sistemas que detallan las otras cuatro partes.

Pese a que el resto de partes de la norma IEC 62580 están aún bajo desarrollo, la primera parte de la norma, en relación a los servicios de mantenimiento, introduce las siguientes capacidades y servicios del OMTS (IEC 62580-5):

- Monitorización y diagnosis remota.
- Telemetría.
- Acceso a documentación y manuales de mantenimiento.
- Servicios para gestión de repuestos.

Con la implementación de estos sistemas con estas capacidades se posibilita la gestión integral de la flota de trenes de un país o de una compañía de ferrocarril.

3.4. Industria del automóvil

Teniendo en cuenta que el automóvil es la plataforma multisistema más común en el mundo, es razonable que la industria del automóvil sea una de las más interesadas en desarrollar técnicas mejoradas para el mantenimiento de sus productos.

Hasta hace tres décadas el automóvil era una plataforma formada por diferentes sistemas electromecánicos. Desde entonces, la electrónica se incorporó de forma masiva en la industria del automóvil, revolucionado gran parte de los sistemas que componen un automóvil.

Desde ese momento lo nuevos sistemas electrónicos demandaban un mantenimiento muy diferente al mantenimiento tradicional que habían demandado los automóviles hasta ese momento. Por este motivo junto a la necesidad de minimizar los costes del ciclo de vida del automóvil, los diferentes fabricantes de automóviles junto con los fabricantes de sistemas para el automóvil, comenzaron a desarrollar y equipar los automóviles con sistemas de mantenimiento electrónicos cada vez más complejos, conocidos como *On-Board Diagnostic* (OBD) [14], [15].

Si bien inicialmente las capacidades OBD se incorporaban solamente en los sistemas más complejos del automóvil, como las Unidades de Control del Motor (ECU), hoy en día la mayor parte de los sistemas embarcados en un automóvil integran este tipo de capacidades para el mantenimiento.

Desde el punto de vista de normativas, dependiendo de los fabricantes de automóviles, fabricantes de sistemas del automóvil, países, etc., las diferentes entidades de normalización (ISO, SAE,…), han desarrollado múltiples normas con el fin de estandarizar los sistemas de mantenimiento [16], [17], [18] y [19].

Respecto al estado del arte para sistemas de mantenimiento centralizados para la industria del automóvil, se considera como representativa la norma ISO 14229 - *Road vehicles - Unified diagnostic services* (UDS) [20].

3.4.1. ISO 14229 - *Road vehicles - Unified diagnostic services (UDS)*

La norma ISO 14229 [20] tiene dos objetivos principales:

- Establecer y unificar las diferentes funciones y servicios de diagnóstico y mantenimiento de un automóvil con los equipos de mantenimiento de taller.

- Aplicar estas funciones y servicios a los diferentes protocolos de comunicación utilizados en los sistemas de mantenimiento de abordo (OBD) desarrollados por los fabricantes de automóvil.

Para cumplir con estos dos objetivos, la norma ISO 14229 está dividida en 7 partes.

- ISO 14229-1:

  Detalla la especificación y define los requisitos de los servicios de diagnóstico unificados.

- ISO 14229-2:

  Establece los requisitos para que los servicios que unifica esta norma sean independientes del protocolo que se utilice.

- ISO 14229-3 a -7:

  Unifican los servicios de diagnóstico para los distintos protocolos: CAN, *FlexRay*, *Internet Protocol* (IP), *K-Line* y *Local Interconnect Network* (LIN).

La Tabla I representa en un esquema de capas OSI [21], [22], las capas que ocupan las distintas partes de la norma ISO 14229.

CAPA OSI	**ISO 14229**
7	ISO 14229-1, -3, -4, -5, -6, -7
6	Capa especifica del fabricante
5	ISO 14229-2
4	Otras normas ISO anteriores.
3	
2	
1	

Tabla I. Esquema OSI de la norma ISO 14229.

Si bien la primera edición de la norma ISO 14229-1 fue publicada en 2006, esta norma continúa en desarrollo, con el fin de incluir los nuevos protocolos de comunicación en los sistemas de mantenimiento del automóvil, de hecho la ISO 14229-7 se ha publicado este mismo año.

4 Técnicas de Mantenimiento de Sistemas Complejos

Este capítulo, Técnicas de Mantenimiento de Sistemas Complejos, desarrolla los diferentes tipos de arquitecturas de sistemas, seguido de las diferentes técnicas de mantenimiento de sistemas, el desarrollo del sistema de mantenimiento basado en BITE y las diferentes técnicas de BITE.

4.2. Arquitectura de sistemas

La rápida evolución tecnológica de las últimas décadas ha llevado consigo una revolución en las filosofías utilizadas para el diseño de sistemas y por tanto en las arquitecturas de los sistemas.

Haciendo un recorrido por las distintas arquitecturas de sistemas utilizadas a lo largo del tiempo, y teniendo en cuenta que un sistema es un conjunto de partes interrelacionadas, comprobamos que la evolución de las arquitecturas de sistemas se ha basado principalmente en el concepto de "compartir".

4.2.1. Sistemas independientes

Inicialmente, los sistemas se diseñaban basados en una arquitectura independiente. En este tipo de arquitectura, cada sistema está diseñado para realizar una o varias funciones específicas. La Figura 1 muestra como ejemplo, el diagrama de bloques de un Sistema Patrón independiente. En este ejemplo, el sistema independiente, está formado por una caja principal en la cual residen las funciones del sistema, y conectado con diferentes equipos periféricos, como el sensor, el actuador y los controles e indicadores.

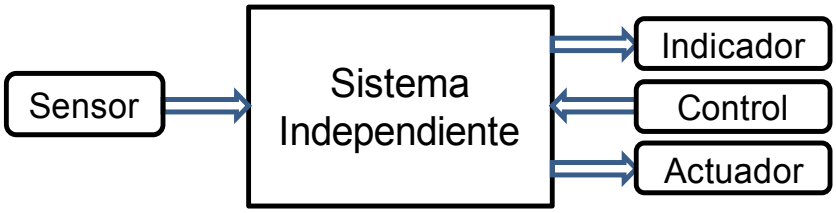


Figura 1. Ejemplo de sistema independiente.

En este tipo de arquitectura, los sistemas no comparten la información ni los recursos del propio sistema, con otros sistemas. De este modo los diferentes sistemas de una plataforma serán totalmente independientes unos de otros.

Las principales ventajas que tienen este las arquitecturas de sistemas independientes:

- Segregación de funciones:

  Las funciones de cada sistema no dependen o influyen en las funciones de otros sistemas, garantizando su independencia.

- Acceso al procesador y a I/O garantizado:

  Los recursos hardware del sistema como el procesador módulo entrada/salida, interfaces, no se comparten con otros sistemas.

- Fácil implementación del sistema:

  Debido a que este tipo de sistemas carece de interfaces con otros sistemas, la implementación de es más sencilla con respecto a otras arquitecturas, mostradas en la Figura 2 y la Figura 3.

Por el contrario, este tipo de arquitectura también tiene desventajas como:

- Uso ineficiente de recursos:

  Debido a que en este tipo de arquitecturas cada sistema dispone de sus propios recursos hardware como fuente alimentación, procesador, módulos entrada salida, en una plataforma con diferentes sistemas, todos estos recursos están redundados.

- Incremento del peso, consumo de potencia y cableado:

  Provocado por la no optimización de los recursos hardware. En algunas plataformas multisistema, como por ejemplo un avión, estos factores son esenciales.

- Certificación al mismo nivel:

  Todas las funciones de un sistema, al estar en el mismo equipo, están certificadas con el mismo nivel, sin tener en cuenta su criticidad.

- Diferentes sistemas de mantenimiento:

  Cada sistema independiente puede tener sistemas de mantenimiento diferentes, incrementando el tiempo y el coste del mantenimiento y aislamiento de fallos en los sistemas.

4.2.2. Sistemas integrados

La arquitectura de sistemas integrados surge al aplicar el concepto de "compartir" sobre la Arquitectura de Sistemas Independientes.

Al analizar una plataforma multisistema formada por diferentes Sistemas Independientes, ver Figura 2, se puede observar que muchas de las partes o módulos que forman estos sistemas desempeñan la misma función. Ejemplos de estas partes análogas, son: los módulos de alimentación, las unidades de control y presentación de información, los módulos de gestión de entrada salida.

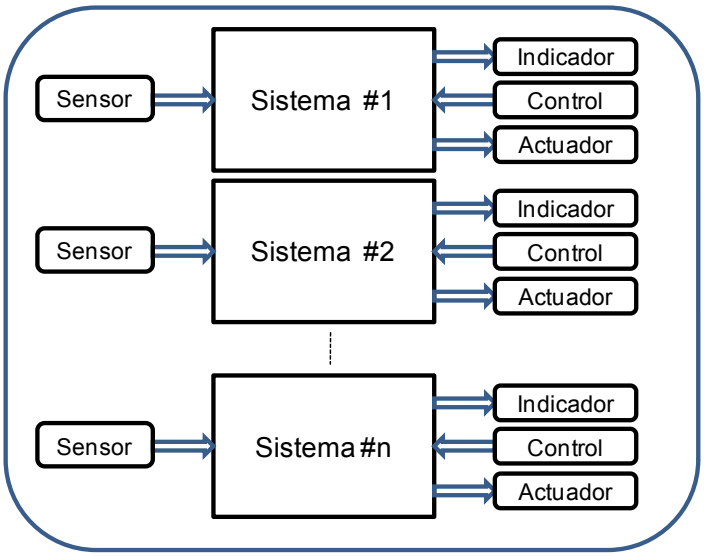


Figura 2. Plataforma con múltiples Sistemas Independientes.

Tras realizar el análisis de los diferentes sistemas independientes, y una vez identificadas las sinergias entre sistemas, se puede aplicar el concepto de "compartir" sobre los diferentes elementos de estos sistemas. De este modo, se puede llegar a establecer la Arquitectura de Sistema Integrados, como muestra la Figura 3.

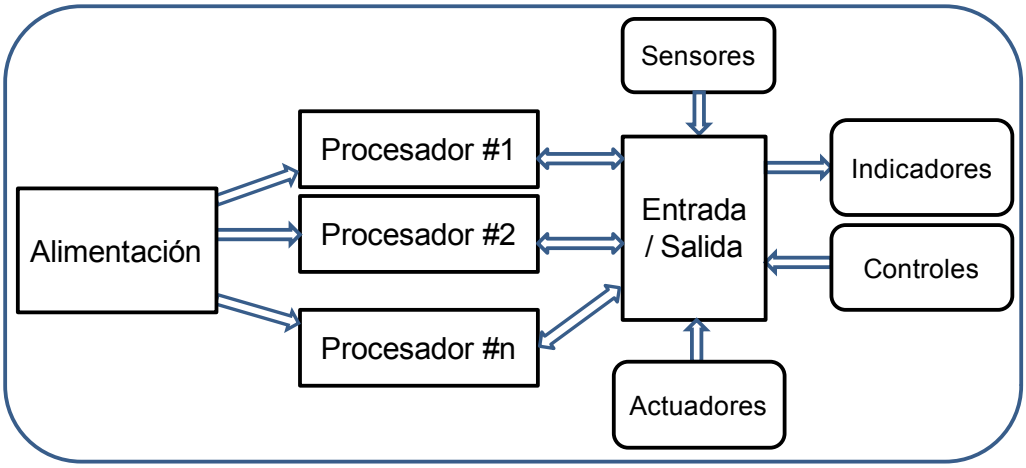


Figura 3. Sistema integrado.

La Figura 3 muestra el uso compartido de algunos módulos por los diferentes sistemas. La implementación de una arquitectura de sistemas integrados normalmente se realiza sobre un rack en el cual se encuentran empotradas diferentes tarjetas, correspondientes a los diferentes módulos que lo forman: Alimentación, Procesadores, Módulos de Entrada / Salida, etc.

Las principales ventajas que tienen frente a las arquitecturas de sistemas independientes:

- Optimización de recursos, reducción de peso, menor consumo energético. En algunas plataformas como un avión, estos factores son determinantes.
- Segregación de funciones según criticidades.
- Gestión de las múltiples aplicaciones centralizada. En este tipo de sistemas las funciones de BITE de cada componente están gestionadas de forma centralizada en una de los componentes del sistema.

Por el contrario, este tipo de arquitectura también tiene desventajas como:

- Complejidad en las interfaces:

  La mayor integración entre los sistemas y la reducción de partes redundantes hacen necesario el uso compartido de interfaces. En consecuencia, el intercambio de información entre equipos o sistemas a través de una interfaz compartida, requiere de un diseño más complejo que con sistemas independientes.

- Integración de los sistemas compleja:

  Al igual que ocurre con las interfaces, la integración de los sistemas en una plataforma aumenta de complejidad respecto a los sistemas independientes.

- Fallos multisistema:

  El hecho de compartir recursos entre los diferentes sistemas, puede provocar que un fallo en uno de estos recursos compartidos afecte a varios sistemas.

- Evoluciones complejas:

  En este tipo de arquitecturas, la evolución de uno de los sistemas implica la necesidad de mantener la compatibilidad con el resto sistemas, haciendo costosos los procesos de evolución y mejora de los mismos.

4.2.3. Sistema Modular Integrado

Al realizar un análisis en profundidad de la arquitectura de sistemas integrados, se puede observar que los diferentes sistemas que forman parte de esta arquitectura, están integrados en un hardware que en muchas ocasiones es similar.

En consecuencia, si el hardware de diferentes sistemas es similar, se puede llegar a concluir que la „esencia" de un sistema reside en su funcionalidad.

Partiendo de estas dos ideas, y con el soporte de la evolución de la tecnología, se puede plantear una arquitectura de sistemas modulares integrados. Este tipo de arquitectura requiere la integración de diferentes aplicaciones sobre un mismo hardware. Estas aplicaciones son las encargadas de desarrollar las funciones que en otro tipo de arquitecturas, desarrollaban los diferentes sistemas.

De forma gráfica la Figura 4 muestra el cambio de filosofía entre sistemas convencionales, y un Sistema Modular Integrado.

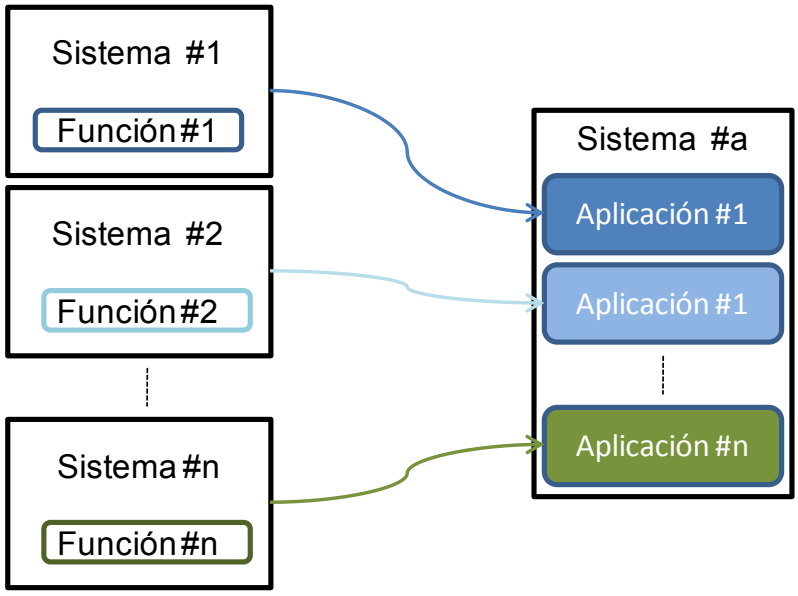


Figura 4. Evolución a un Sistema Modular Integrado.

En la industria aeronáutica las arquitecturas de sistemas modulares integrados, están teniendo una gran aceptación en los desarrollos de nuevos aviones [23], [24].

Con el fin de definir y estandarizar estas arquitecturas para la industria de aeronáutica, la entidad RTCA ha definido la norma DO-297 Aviónica Modular Integrada (IMA) [24] y la entidad ARINC ha definido la ARINC 653 *Avionics Application Standard Software Interface* [25].

La norma DO-297 define el concepto IMA como "*un conjunto compartido de recursos flexibles, reutilizables, e interoperables formados por hardware y software que integrados forman una plataforma que proporciona servicios,*

diseñados y verificados contra un conjunto de requisitos de capacidades y de seguridad, lo cual permite alojar aplicaciones que realizan las funciones de las aeronaves".

La norma ARINC 653 establece una especificación de software que debe gestionar todos los recursos de espacio y tiempo del sistema entre las diferentes aplicaciones de tiempo real, incluyendo espacio de memoria y tiempo de CPU asignado a cada una de las aplicaciones.

Para los sistemas integrados bajo este tipo de arquitectura, la gestión del mantenimiento basado en las funciones de BITE de los diferentes equipos, se realiza desde un Sistema de Mantenimiento Centralizado. Este es un sistema específico para mantenimiento el cual que se encarga de las siguientes funciones:

- Gestión y almacenamiento de datos de fallos.
- Ejecución de test de diagnóstico.
- Ejecución de test funcionales.
- Asistencia en la localización de averías.

El Sistema de Mantenimiento Centralizado se desarrolla en el capítulo 5.

Las principales ventajas que presenta la arquitectura de sistemas modulares e integrados, son:

- Sistema escalable, abierto y modular:

  Estas tres características, permiten que la integración de nuevas aplicaciones en la plataforma sin impactar en las prestaciones del resto de aplicaciones, bien permitiendo integrar una nueva aplicación junto a otras en un hardware existente, o bien incluyendo un nuevo equipo hardware en la plataforma.

- Diferentes funciones con diferentes niveles de criticidad:

  Sobre el mismo equipo, pueden estar alojadas funciones o aplicaciones desarrolladas con diferentes niveles de criticidad (DAL A, B, C, D o E) [26].

- Portabilidad del software:

  El software que define el sistema empotrado o aplicación, es independiente del equipo en el que está alojada. Por lo tanto, podremos cambiar o actualizar dicha aplicación, sin tener que cambiar el equipo que la aloja.

- Independencia de sistemas:

  Aunque sobre un mismo equipo hardware residan diferentes sistemas o aplicaciones, el diseño, desarrollo, pruebas e integración, de estos sistemas o aplicaciones son independientes entre sí. De este modo se reduce la necesidad de realizar pruebas de integración conjunta de las distintas aplicaciones.

- Cualificación y certificación:

  Al igual que para el diseño, desarrollo y pruebas, las aplicaciones son independientes, ocurre lo mismo para las actividades de calificación y certificación, que pueden realizarse por separado para las distintas aplicaciones.

- Arquitecturas estandarizadas:

  El desarrollo de este tipo de arquitectura de sistemas está definido por las normas ARINC 653 [25] y DO-297 [24].

- Mantenimiento eficaz:

  La gestión del mantenimiento de estas arquitecturas se realiza desde un Sistema de Mantenimiento Centralizado.

- Mínimos riesgos por obsolescencia:

  Los sistemas desarrollados son independientes del soporte hardware en la cual están alojados, por lo que la obsolescencia de alguno de estos soportes hardware no suponen un riesgo para el mantenimiento operacional de los sistemas.

- Reduce el peso, volumen, los costes, el tiempo de desarrollo y el consumo de energía.

Por el contrario, la arquitectura de sistemas modulares integrados, presenta una serie de inconvenientes:

- Complejidad del desarrollo de los sistemas:

  Incrementa las actividades de diseño, desarrollo y producción, por lo tanto incrementa los costes.

- Procesos de certificación costosos:

  Debido a la novedad de este tipo de arquitecturas, los procesos de certificación aún no están lo suficientemente maduros y conllevan un mayor esfuerzo de lo esperado inicialmente.

- Número limitado de suministradores:

  Al igual que en el punto anterior, la novedad de este tipo de arquitecturas hacen que aún el número de fabricantes de este tipo de sistemas sea reducido y los costes elevados.

- Arquitecturas recientes:

  Este tipo de sistemas sólo se han instalado en las últimas generaciones de aviones como en los Airbus A380, A400M y el Boeing 787.

En las secciones anteriores se han revisado las diferentes tendencias en las arquitecturas de sistemas. A continuación se presentan las principales modalidades y tendencias en el mantenimiento de sistemas.

4.3. Principales técnicas y métodos de Mantenimiento de sistemas

Desde la revolución industrial el concepto de mantenimiento de sistemas ha evolucionado según se ha incrementado la complejidad de los mismos y las diferentes tecnologías utilizadas.

Si bien el concepto de la palabra mantenimiento inicialmente puede parecer relativamente sencillo, en la actualidad, no lo es tanto el definir el

alcance de este concepto. Entre las múltiples definiciones existentes para mantenimiento, la norma EN 13306 [27] define mantenimiento como: *"Combinación de todas las acciones técnicas, administrativas y de gestión, durante el ciclo de vida de un elemento, destinadas a conservarlo o devolverlo a un estado en el cual pueda desarrollar la función requerida".*

La Figura 5 muestra los tres principales tipos de mantenimiento, los cuales están desarrollados en los siguientes apartados.

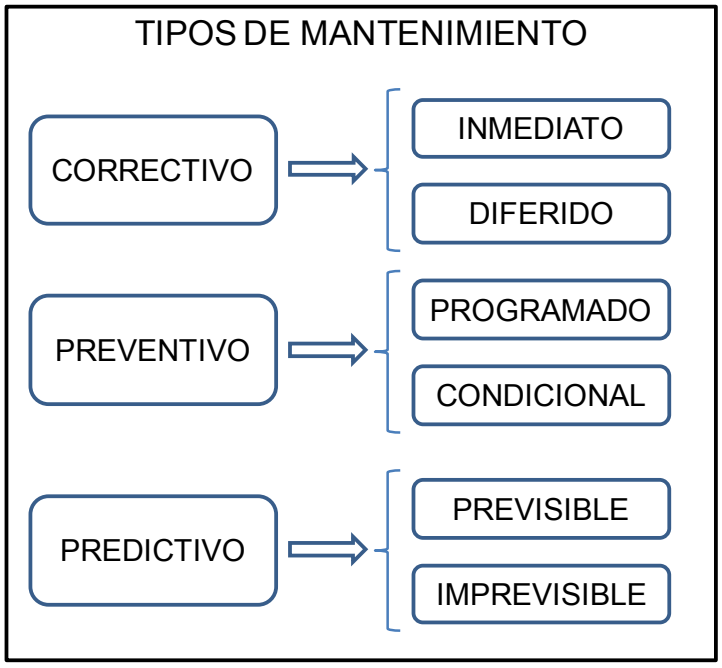


Figura 5. Tipos de Mantenimiento

4.3.1. Mantenimiento correctivo

El mantenimiento correctivo fue el primer método de mantenimiento utilizado, no obstante, hoy en día se sigue utilizando en diferentes sistemas, como por ejemplo sistemas en los que por su baja criticidad pueden ser interrumpidos en cualquier momento o sistemas para los cuales el coste del mantenimiento correctivo sea menor que el del mantenimiento preventivo.

Este método de mantenimiento se basa en aplicar una tarea de mantenimiento una vez que se presenta una condición de fallo en un equipo.

Las tareas aplicadas para el mantenimiento correctivo, consisten en reparaciones y sustituciones de los elementos o equipos dañados, que permiten que el sistema vuelva a estar operativo de acuerdo a su especificación de diseño.

En función de la necesidad de restaurar el sistema su operación tras un fallo, podemos clasificar el mantenimiento correctivo como:

- Diferido:

  Cuando las consecuencias del fallo están acotadas y no suponen un impacto grave en la operación del sistema. De tal forma que el sistema pueda seguir operando de forma temporal con ese fallo, o bien seguir operando en un modo degradado. De este modo el fallo no obliga a una parada inmediata del sistema con el fin de aplicar la tarea de mantenimiento correspondiente.

- Inmediato:

  En este caso, las consecuencias del fallo obligan a detener la operación del sistema y aplicar la tarea de mantenimiento correspondiente.

Según el origen de las tareas de mantenimiento correctivo puede clasificarse como:

- Previsibles:

  Cuando el fallo corresponde a un elemento o equipo el cual ha superado el límite de su vida útil.

- Imprevisibles:

  Cuando el fallo de un equipo o un elemento ha sido provocado por un condicionante externo.

La ventaja principal del mantenimiento correctivo es que se agota la vida útil del equipo. Pese a esta ventaja el mantenimiento correctivo puede presentar los siguientes inconvenientes:

- El fallo puede aparecer en un momento crítico durante la operación del sistema.
- Fallos no detectados a tiempo pueden causar daños irreparables en otros elementos, con la posibilidad de impactar en la seguridad del sistema.
- En necesario disponer de un stock de equipos de repuesto.

4.3.2. Mantenimiento preventivo

El mantenimiento preventivo o también denominado mantenimiento programado tiene como propósito prevenir la aparición de fallos o puesta a punto de los equipos o sistemas.

El diseño de este tipo de mantenimiento está basado en la experiencia acumulada y en la estimación o conocimiento de la vida útil de un equipo con el fin de reemplazarlo o repararlo antes del fin de su vida útil.

Podemos clasificar el mantenimiento preventivo en dos tipos:

- Sistemático:

  Consiste en aplicar las tareas de mantenimiento correspondiente cada cierto número de horas, de acuerdo a las recomendaciones de los fabricantes de los equipos o sistemas.

- Condicional:

  Cuando la aplicación de las tareas de mantenimiento se realiza en función de condicionantes. Estos condicionantes pueden ser indicadores de desgaste o degradación en el equipo, por ejemplo el consumo excesivo de corriente.

Los objetivos principales del mantenimiento programado son minimizar fallos, prolongar vida útil, incrementar eficiencia, calidad, y seguridad en la operación del sistema. Sin dejar a un lado la reducción de costes por actividades de mantenimiento no programadas y las consecuencias derivadas de paradas no programadas del sistema.

Para conseguir estos objetivos el mantenimiento preventivo se basa principalmente en dos tipos de actividades:

- Inspecciones periódicas, correspondientes al mantenimiento preventivo sistemático.
- Acciones de mantenimiento no programadas, correspondientes al mantenimiento preventivo condicional.

La aplicación del mantenimiento preventivo en un sistema o en una plataforma se debe realizar de acuerdo a un plan de mantenimiento programado, ver apartado 4.3.2.1.

La ventaja principal que presenta el mantenimiento preventivo, es que llevando a cabo el plan de mantenimiento programado de un sistema o plataforma, se consigue mantener los sistemas en un estado óptimo de funcionamiento, detectando posibles fallos durante las inspecciones periódicas y reemplazando los equipos antes de que finalicen su vida útil. De este modo se minimiza las consecuencias derivadas de los fallos en los sistemas. Asimismo permite planificar los recursos necesarios para el desarrollo de las tareas de mantenimiento programadas.

Por el contrario el mantenimiento programado también presenta algunos inconvenientes, como por ejemplo:

- Rentabilidad:

  Los equipos se reemplazan sin haber agotado su vida útil.

- No se elimina la probabilidad de una avería inesperada:

  Los tiempos estimados de vida de los sistemas, se calculan de forma estadística, por lo que en ocasiones, los fallos de un equipo o sistema se darán antes de la fecha de vida limite estimada.

- Dificultad en el ajuste temporal de las diferentes tareas de mantenimiento.

4.3.2.1. Mantenimiento programado en aeronaves (MSG3)

En la industria aeronáutica, la evolución tecnológica y el aumento de la complejidad de los sistemas han provocado que el mantenimiento de las aeronaves sea cada vez era más costoso y complejo.

Con el fin de mejorar los procesos de mantenimiento de los aviones el organismo *Air Transport Association* (ATA), definió el *Maintenance Steering Group* (MSG). Este organismo desarrolló inicialmente el método de mantenimiento MSG1, y tras varias evoluciones ha llegado a establecer el método MSG3 [4].

El método MSG3 principalmente se desarrolló en torno a dos propósitos del anterior método MSG2:

- Separar los aspectos relativos a seguridad de los aspectos económicos.
- Definir procedimientos para detectar los fallos ocultos.

Basándose en estos dos propósitos, el MSG3 define un proceso lógico con el cual se determinan que acciones son necesarias para asegurar la disponibilidad de un recurso dentro de su operación específica.

Una característica importante del MSG3 es que las tareas de mantenimiento requeridas son a nivel sistema y no a nivel componente como ocurría anteriormente con el MSG2. Esto se puede traducir en que si demuestra que un fallo funcional de un sistema no afecta a seguridad y adicionalmente no tiene un impacto económico, no tiene por qué requerir una tarea de mantenimiento programada.

El resultado del MSG3 es un Plan De Mantenimiento Programado (MRB).

El proceso MSG3 está dividido en cuatro secciones:

- Sistemas y plantas de potencia.
- Estructuras.
- Inspecciones zonales.

- Rayos y fuentes radiantes de alta intensidad.

Una ventaja del MSG3 es que mejora los estándares de seguridad, dado que se optimizan las tareas de mantenimiento evitando tareas innecesarias.

Hoy en día los procesos MSG3 están aceptados por las diferentes autoridades de aviación de los principales países, los fabricantes de aeronaves y por los operadores, en tal grado que la autoridad obliga a seguir los procesos MSG3 para el diseño del plan de mantenimiento programado de las aeronaves. Un ejemplo de esto es la *Advisory Circular* AC-121-22A de la *Federal Aviation Administration*, (FAA) [28]. Este documento proporciona una guía a la industria de la aviación, para revisar los planes de mantenimiento según las recomendaciones de la FAA, basándose en el MSG3.

El resto de las tareas de manteamiento de una aeronave que no están incluidas en el MRB, son aquellas tareas necesarias para corregir discrepancias o defectos observados durante las inspecciones de las tareas de manteamiento programadas.

4.3.3. Mantenimiento predictivo

Dependiendo de la bibliografía, el mantenimiento predictivo, también puede ser denominado monitorización en línea, mantenimiento basado en condiciones o mantenimiento basado en riesgos [5].

El mantenimiento predictivo surge a partir del mantenimiento preventivo en respuesta a las limitaciones que presenta.

El mantenimiento predictivo se define como la toma de acciones correctoras en base a condiciones. Dichas condiciones deben relacionar variables físicas del equipo o sistema con su estado.

Aunque la definición del mantenimiento predictivo es sencilla, la clave para el éxito de este tipo de mantenimiento está en establecer correctamente las condiciones que evalúan el sistema.

La evaluación de las condiciones se realiza mediante la monitorización de los parámetros característicos de los equipos o sistemas durante su la operación.

Como parte del mantenimiento predictivo, se debe definir las acciones de mantenimiento en función de la criticidad del fallo. De este modo, dependiendo del sistema se pueden llevar a cabo distintas acciones antes de detener el funcionamiento del sistema, para la sustitución de la unidad fallada. Ejemplos de estas acciones son la reconfiguración del sistema o continuar la operación del sistema con limitaciones.

El principal beneficio del mantenimiento predictivo es la anticipación al fallo de un equipo. Lo cual permite aproximarse al límite de la vida útil de los equipos o sistemas, sin llegar a que se produzca el fallo de los mismos, de esta forma se rentabilizan al máximo los sistemas sin incurrir en la problemática del mantenimiento correctivo.

Por el contrario, este tipo de mantenimiento también presenta algunos inconvenientes como:

- No todas las causas de fallo de un equipo o sistema pueden ser detectadas con antelación.
- La monitorización de los equipos y sistemas incrementan su coste.

A pesar de los avances en las tecnologías de mantenimiento predictivo, las técnicas de mantenimiento preventivo mediante inspecciones periódicas aún son técnicas muy utilizadas en muchos de los procesos industriales.

Como ejemplo particular del método de mantenimiento preventivo mediante inspecciones periódicas, son las calibraciones anuales de equipos, las cuales son requeridas por los procesos de calidad y/o por las normativas.

4.3.4. Mantenimiento proactivo

La filosofía del mantenimiento proactivo consiste en realimentar el diseño y desarrollo de un sistema con la experiencia de los fallos que este presenta a lo largo de su vida operativa, con el fin de evitar y corregir el origen de los mismos.

De este modo se consiguen sistemas con una elevada fiabilidad con los cuales se minimiza los costes de mantenimiento.

Los principales problemas que presenta el desarrollo de sistemas con mantenimiento proactivo son los largos procesos de desarrollo y los elevados costes que llevan consigo, siendo necesario desarrollar varias versiones de los sistemas, hasta conseguir el sistema final.

Esta metodología es utilizada por los fabricantes cuando continúan evolucionando un sistema, a lo largo de su ciclo de vida.

4.4. Mantenimiento basado en *Built-In-Test Equipment* (BITE)

Desde hace algunos años se están desarrollando ciertos sistemas los cuales integran funciones de mantenimiento basadas en la monitorización del propio sistema.

Para la integración de estas funciones de mantenimiento es necesario incluir en el sistema un hardware y/o software adicional, sobre los cuales se puedan implementar funciones de monitorización, con el fin de ofrecer la capacidad de detección y aislamiento de fallos para el sistema.

El conjunto de hardware, software y funciones de mantenimiento de un sistema, se denominan *Built-in-test-equipment* o BITE.

Consecuentemente, los requisitos para la implementación de las capacidades de BITE en un sistema, se deben tener en cuenta desde el inicio del diseño de dicho sistema [29], [30].

Pese a la definición del BITE, durante años este término se ha utilizado de forma inexacta, bien para denominar distintas tareas o funciones de sistemas

como pruebas funcionales, o para referenciar la gestión de alertas y avisos, incluso se ha utilizado únicamente para denominar al hardware de mantenimiento de un equipo o incluso para señalar el pulsador de test de un equipo.

Los objetivos principales que debe alcanzar la función BITE de un sistema es:

- Conseguir una cobertura completa de los fallos que puede presentar un sistema.
- Analizar los fallos que sucedan en el propio sistema con el fin de aislar el equipo fallado.
- Gestionar los test del sistema.
- Gestionar y memorizar los datos los datos de fallos para reportar los equipos fallados.

4.4.1. Detección y aislamiento de fallos

Para la implementación de las funciones BITE, se utilizan principalmente dos técnicas de detección de fallos:

4.4.1.1. Monitorización

La monitorización es una técnica de BITE no intrusiva para el sistema bajo análisis, debido a que no tiene capacidad para inducir posibles daños en el sistema. La monitorización está basada en la comparación de un parámetro característico del sistema con un patrón de ese parámetro. Este parámetro puede ser un patrón físico o bien un modelo de dicho parámetro. Desde el punto de vista del sistema, la monitorización es considerada una aplicación operacional del propio sistema.

La clave fundamental para el buen funcionamiento de la monitorización es la correcta selección de los umbrales de detección y el retardo de las señales patrón utilizadas para comparar los parámetros del sistema.

4.4.1.2. Test

El test es una técnica de BITE considerada una técnica intrusiva. La técnica de test está basada en la estimulación de una parte del equipo, con el fin de evaluar la respuesta del equipo a dicho estimulo, y poder detectar posibles fallos en el equipo. Dependiendo del modo de funcionamiento, esta técnica de BITE puede ser activada automáticamente o de forma manual.

Por lo tanto, el resultado de la función BITE, es un mensaje de fallo.

Cuando a través de la monitorización o de los test, se detecta una condición de fallo en el sistema, la función BITE, recibe la confirmación de la condición de fallo, junto con el estado de ciertos parámetros característicos del sistema. De este modo la función BITE analiza la condición de fallo junto a las condiciones del sistema con el fin de generar el mensaje de fallo correcto que acuse al equipo que ha originado el fallo.

Si bien la mayoría de las condiciones de fallo de un sistema están directamente asociadas al equipo que origina ese fallo, en ciertas ocasiones, es necesario recurrir al análisis en profundidad de los parámetros característicos del sistema monitorizados en el momento del fallo.

4.4.1.3. Funciones BITE

La capacidad BITE de un sistema dispone de tres modos de funcionamiento:

- *Power On Self Test* o Test de Inicio:

  El test de inicio, se ejecuta de forma automática, al energizar el sistema. El objetivo principal de este test es garantizar que el sistema opera de modo seguro.

  Para cumplir este objetivo, durante el test de inicio, se ejecutan pruebas relacionadas con el hardware y el software, que puedan detectar fallos latentes en el sistema, los cuales puedan llegar a tener un impacto en la seguridad del sistema o de la plataforma donde esté instalado. De esta forma, si durante el test de inicio no se reporta

ningún mensaje de fallo, se entiende que quedan garantizadas las principales condiciones operacionales del sistema.

La duración del test de inicio, debe estar limitada, en función de los requisitos de la plataforma, en la que vaya embarcado el sistema.

La ejecución del test de inicio a parte de ejecutarse cuando se inicia el sistema, también debe ejecutarse tras un reinicio del sistema.

En caso de que el test de inicio detectara algún fallo en el sistema, el BITE del sistema debe propagar el mensaje de fallo correspondiente al fallo detectado, a través de la interfaz del BITE, y en paralelo, el sistema activará el respectivo aviso de fallo, a través del panel de control del sistema, o del sistema de avisos de la plataforma, con el fin de alertar al operador del sistema.

El resultado del test de inicio, junto con los parámetros característicos del sistema, debe ser almacenado en una memoria no volátil del sistema.

- Monitorización Continua:

  Tras finalizar el test de inicio, y de forma periódica la función BITE monitoriza el correcto funcionamiento del sistema durante el tiempo que el sistema está en operación.

  Un requisito imprescindible de la monitorización continua, no debe tener ningún efecto en la operación del sistema.

  Durante la monitorización continua del sistema, la función BITE, es la encargada de comprobar el correcto funcionamiento de:

  - Fuentes de alimentación,
  - Módulos de entrada / salida,
  - Buses de comunicaciones,
  - Señales discretas, analógicas y digitales,
  - Circuitos digitales

- Software.

- Test Interactivo:

  La tercera función del BITE es el test interactivo, el cual permite al operador de mantenimiento del sistema, la ejecución de este test cuando sea necesario. La principal función del test interactivo es confirmar si el sistema presenta algún fallo o si por el contrario el sistema funciona correctamente.

  El comando del test interactivo a un sistema, se realiza en función de las capacidades de las interfaces del BITE de dicho sistema. En el apartado 4.4.1.4, se desarrollan las distintas interfaces para BITE de un sistema.

  El test interactivo no debe afectar a los equipos que tienen interfaz con el sistema bajo test, por lo tanto la ejecución de este tipo de test no provocará ningún efecto en las entradas o salidas del sistema.

  En función de la complejidad del sistema, la función BITE puede disponer de diferentes tipos de test interactivos, con diferentes niveles de profundidad.

4.4.1.4. Tipos de interfaces del BITE

En función de las capacidades de comunicación de la función BITE de un sistema podemos encontrar tres tipos de BITE:

- Señal discreta:

  Los primeros equipos que incorporaron capacidades BITE, únicamente, indicaban su estado a través de una señal discreta. Con la activación de esa señal, el sistema indicaba si había detectado algún fallo. Adicionalmente, de forma interna, el sistema podía registrar el fallo que había detectado.

  La activación del Test de Inicio podía realizarse a través de algún pulsador del propio equipo o de su panel de control.

- Bus unidireccional y señal discreta:

  Algunos equipos disponen de un bus unidireccional más una señal discreta como interfaces para la función BITE. En estos sistemas, la función BITE utiliza el bus para propagar la información de los mensajes de fallo. El destinatario de esta información será un equipo de mantenimiento externo. Por otro lado, a través de una señal discreta, el sistema puede recibir el comando de un test interactivo.

- Bus bidireccional:

  Los últimos sistemas desarrollados, incorporan funciones BITE con conexión a través de bus bidireccionales. El propósito de este bus de comunicaciones bidireccional para BITE, es enviar la información de los mensajes de fallo generados por la función BITE, a otro sistema de mantenimiento externo, y recibir comandos para la realización de los test interactivos desde el sistema de mantenimiento externo.

4.4.1.5. Mensajes de fallo de BITE

En la definición de los mensajes de fallo de generados por la función BITE para un sistema deben tener en cuenta una serie de requisitos básicos:

- Un mensaje de BITE debe de indicar el origen del fallo a nivel equipo, o interfaces, pero no debe indicar fallos internos del equipo.

  Este requisito, está basado en el tipo de mantenimiento a realizar sobre la plataforma en la que está instalado el sistema en cuestión. El mantenimiento típico a realizar en una plataforma, en caso de que un equipo presente un fallo, es aplicar la tarea de mantenimiento para la su sustitución de dicho equipo. Por lo tanto, las reparaciones internas de un equipo quedan fuera de los objetivos del BITE.

Adicionalmente, el propio equipo internamente puede almacenar información sobre la condición de fallo sufrida. Esta información es de utilidad para llevar a cabo la reparación del equipo por el centro reparador correspondiente.

- Un mensaje de BITE debe de indicar fallos externos al equipo como fallos en las interfaces del equipo, o fallos ocasionados por sistemas no conectados.

  La función BITE de un sistema tiene capacidad de detectar un fallo en las interfaces del sistema, por lo tanto en caso de producirse una condición de fallo en una de las interfaces, la función BITE debe propagar el mensaje de fallo correspondiente y acusar del origen del fallo a las posibles causas: interfaz y equipo o equipos conectados a esa interfaz.

- Un mensaje de fallo de BITE debe requerir una única acción de mantenimiento.

  Dado que el BITE debe facilitar el mantenimiento de los sistemas, un mensaje de BITE debe ser claro y conciso, evitando que el operario de mantenimiento pueda dudar en la tarea de mantenimiento que debe aplicar para restaurar la correcta operación del sistema fallado.

4.4.1.6. Datos de ingeniería

Algunos sistemas que disponen de capacidades de BITE, a parte de las funciones desarrolladas anteriormente, disponen de la capacidad de registrar una serie de datos característicos del sistema en torno al instante en que se produce un fallo.

Esta información se almacena con el objetivo de proporcionar información adicional sobre los fallos que presente el sistema, para el fabricante o reparador del sistema y no para el operario de mantenimiento del sistema.

4.4.2. Diseño para la testabilidad (DFT)

Aunque el diseño para testabilidad abarca un amplio campo de posibilidades, podemos definirlo como el conjunto de técnicas a aplicar a un sistema mejorar su correcto funcionamiento [31]. El concepto principal de este tipo de diseño reside en que desde las fases iníciales del diseño se debe disponer de los requisitos de comprobación del sistema a diseñar.

El diseño para la testabilidad se divide en dos tipos:

- Ac-hoc:

  Utilizan soluciones hechas a medida para cada circuito o sistema. El DTF desarrollado con técnicas Ac-hoc, se basa en la aplicación de un conjunto de reglas de diseño desarrolladas con el fin de mejorar el acceso a los nodos internos de un circuito los cuales permiten monitorizar el correcto funcionamiento interno del circuito.

- Estructurado:

  Abordan el problema de la comprobabilidad de un circuito aplicando una serie de normas al diseño. La aplicación de este tipo de diseño incrementa el coste del producto. Las dos principales metodologías son:

  - Scan-path:

    Esta metodología está basada en la aplicación de ciertas normas de diseño que facilitan el correcto funcionamiento interno de un circuito. Estas normas requieren la utilización de circuitería adicional.

    La implementación de esta metodología requiere separar la comprobación de la circuitería secuencial, de la circuitería combinacional. Utilizando vectores de test para la comprobación de la parte combinacional y circuitería adicional como multiplexores y registros de desplazamiento para comprobar las partes secuenciales de los circuitos.

- Ventajas: Reduce el análisis de la parte combinacional a un registro de desplazamiento y permite una aproximación a la detección del origen del fallo.

- Inconvenientes: Muchos ciclos para test, necesita circuitería y pines adicionales, consume recursos del circuito.

- *Built in self-test* (BIST):

  Se basan en la aplicación una serie de vectores o patrones de test sobre el circuito o sistema a analizar, y en el análisis de las respuestas del circuito o sistema ante esos patrones de test, ver apartado 4.4.6.

4.4.3. Técnicas BITE para hardware

4.4.3.1. Monitorización para dispositivos analógicos

A continuación se describen algunos de los principales métodos de monitorización para señales analógicas que se implementan para soportar la funcionalidad BITE de los sistemas.

- Sensores de alimentación:

  La monitorización de las alimentaciones de un sistema es uno de los puntos fundamentales del BITE. En las alimentaciones de corriente continua se monitoriza que los niveles de tensión y el rizado de esta están dentro de los límites y que los transitorios de encendido y apagado no introduzcan picos de tensión y/o corriente. Para las alimentaciones en tensión alterna, además de monitorizar los valores de tensión alterna, se monitoriza su frecuencia.

- Bloqueo de señales discretas:

  Monitorización de líneas discretas con el fin de comprobar que las señales no se quedan bloqueadas en un estado, como por ejemplo un cortocircuito a masa.

- Comparación:

  Si el sistema dispone de señales duplicadas, se podrá comprobar que no existen discrepantes entre ellas. Si es posible, se podrá comparar los valores de un mismo parámetro de fuentes o sensores diferentes. Como ejemplo, un sistema puede recibir un parámetro de velocidad de distintas fuentes: sensor de velocidad y derivada de la señal de posición de GPS.

- Credibilidad:

  Consiste en evaluaciones lógicas entre parámetros o condiciones. Como ejemplo, las señales de monitorización de un servo no pueden indicar a la vez que el servo se está desplazando y tiene activo un final de carrera.

- Rango:

  Monitorización de que los valores de las señales de sensores no superaran los límites del rango definido para ese sensor teniendo en cuenta un porcentaje de tolerancia.

- Realimentación:

  Monitorización de que los valores de las señales retorno, en circuitos de lazos de control están dentro de valores lógicos para el circuito de control en cuestión.

- Continuidad:

  Monitorización de que las variaciones de una señal analógica no presentan saltos o variaciones fuera de sus límites operacionales.

- Linealidad:

  La monitorización de la derivada del valor de una señal detecta no linealidad si deja de ser un valor constante.

- Potenciómetros:

  La monitorización de la tensión en los terminales de un potenciómetro puede detectar una rotura del potenciómetro.

- Resistencias variables:

  Monitorización por comparación de medidas en puentes de Weastone o mediante medidas diferenciales.

- Conmutador:

  La monitorización de la posición del conmutador indicara condición de fallo si detecta ambas salidas en circuito abierto o ambas salidas del conmutador en cortocircuito.

- Motores y sincros resolver:

  Monitorización de la fases de señales de los bobinados de un motor o un sincro, teniendo en cuenta que la suma de las fases debe ser 0.

- Bobinas y condensadores:

  Monitorización por comparación de medidas en puentes de Maxwell o mediante medidas diferenciales.

4.4.3.2. Monitorización de buses

La monitorización de los buses de comunicaciones debe realizarse de acuerdo con las recomendaciones de cada tipo de bus. Normalmente, la monitorización de un bus se realiza de forma periódica por cada sistema conectado a dicho bus. Los buses deben monitorizarse aunque se trate de un bus de reserva o esté en stand-by.

Algunas técnicas de monitorización de buses utilizan:

- Bit de paridad:

  Muchos buses cuentan con un bit de paridad en las tramas de la información transmitida. A través de este bit se puede confirmar si ha habido una posible modificación en la información recibida.

- Frecuencia de refresco:

  En caso de que el bus no se refresque con su periodicidad definida, será considerado un fallo en el bus.

- Métodos propios de cada tipo de bus:

  Por ejemplo el bus A429, dispone de dos bits por palabra denominados SSM, cuyo valor indica el estado del bus, y en función del tipo de información transmitida, puede indicar si el bus esta fallado.

Si un sistema detecta un fallo en un bus, el sistema debe desactivar dicho bus, propagar el mensaje de fallo asociado y almacenar los datos característicos del sistema para ese fallo, además activará el bus de reserva, siempre que esté disponible.

4.4.4. Técnicas BITE para software

En el diseño y desarrollo de aplicaciones software para sistemas empotrados, se deben tener en cuenta desde el inicio del diseño, las diferentes técnicas de diseño aplicables, junto a los requisitos de seguridad y comprobabilidad, las estrategias de operación del sistema y las restricciones que imponga el propio diseño.

Las diferentes técnicas utilizadas para BITE en software tienen el objetivo de detectar posibles errores o fallos en el software que puedan llegar a provocar una con condición de fallo en el sistema.

En función de la criticidad del sistema en el que está empotrado el software, y por lo tanto de las consecuencias que puede tener para el sistema o

para la plataforma la condición de fallo más grave provocada por un fallo del software, se define el nivel de seguridad del software, (DAL) según la norma DO178C [26].

En los sistemas que disponen de un software integrado, desde el punto de vista de comprobación de fallos o test, no solo se debe tener en cuenta el correcto funcionamiento del software, sino también el correcto funcionamiento de la integración software y hardware.

Desde el punto de vista de seguridad y comprobabilidad de la aplicación software se pueden implementar técnicas de diseño como:

- Particionamiento:

  El objetivo de esta técnica es proporcionar aislamiento entre diferentes componentes software. De este modo, un posible fallo en un componente software, queda aislado en dicho componente software, no contaminando al resto de software y permitiendo un aislamiento del fallo más eficaz.

  Estas ventajas se incrementan, si en el particionamiento además de aislar los componentes software, se complementa con la asignación de recursos hardware únicos a cada componente software, consiguiendo aislamiento software y hardware.

  Esta técnica es utilizada para conseguir independencia entre distintas aplicaciones software que comparten un mismo computador, como en los sistemas IMA [24].

  Para conseguir la independencia entre componentes o aplicaciones software, es imprescindible que la implementación de esta técnica garantice que dicho componente o aplicación software:

  - No debe exceder el tiempo ni la capacidad de procesador asignada.
  - No contamine otros componentes o aplicaciones software, ni entradas / salidas, ni áreas de memoria no asignadas.

- Un fallo de en un componente o aplicación software o en un hardware de uso único, no deben influir en otros componentes o aplicaciones software.

- Software Multi-Versión:

  Esta técnica tiene como objetivo el desarrollo de diferentes componentes software los cuales desarrollan la misma función. De este modo, con la implementación de esta técnica se consigue evitar posibles fuentes comunes de errores en la generación del software.

  La implementación de esta técnica, tiene el problema añadido de que el hardware donde esté empotrado este software, debe ser capaz de gestionar las diferentes versiones de un componente software que realizan la misma función.

  Esto se traduce en un mayor coste tanto del proceso de diseño y desarrollo software, como de recursos hardware, provocando que la implementación de esta técnica queda reducida a aplicaciones con una elevada criticidad.

Para la detección de condiciones de fallo específicas en software, al igual que para el hardware, se recurre a la utilización de técnicas de test y técnicas de monitorización.

La implementación de estas técnicas para detección de condiciones de fallo en software, pueden aplicarse tanto al software como a la integración hardware y software.

4.4.4.1. Técnicas de test software

De acuerdo a la división que establecen algunas normas como la DO178C [26], para los test de comprobación software se pueden clasificar en test para condiciones nominales y test para condiciones extremas o test exhaustivos.

El objetivo de las técnicas de test en software es comprobar que el software opera y desarrolla sus funciones correctamente, sin presentar ninguna condición de fallo, en las diferentes condiciones de funcionamiento posibles.

Los test software para condiciones nominales se encargan de comprobar el correcto funcionamiento del software en las condiciones para las cuales ha sido especificado. Las principales comprobaciones que realiza estos tipos de test consisten en:

- Gestión de variables de entrada de tipos entero y real, en los rangos definidos. Una comprobación típica es para los valores límites a los rangos definidos.
- Funciones relacionadas con gestión de tiempos, retardos, bucles, integradores y filtros. Comprobando que no se producen desbordamientos de contadores ni se sobrepasan los tiempos asignados a cada tarea.
- Correcto funcionamiento de todos los modos utilizados en la aplicación software.
- Transiciones entre los diferentes estados de operación nominal.
- Correcto manejo de variables y operaciones lógicas

Los test software exhaustivos son los encargados de comprobar la capacidad del software para gestionar correctamente posibles condiciones para los cuales el software no ha sido especificado. Las comprobaciones características de estos test verifican el adecuado comportamiento del software ante:

- Valores o tipos inválidos para variables de entrada.
- Inicialización de los componentes software en condiciones inválidas.
- Modos de fallo en entradas al sistema.
- Control de contadores en bucles y lazos.
- Tiempos de respuesta de las funciones.

- Protecciones contra desbordamientos de variables o memorias.
- Transiciones no validas entre diferentes estados de la aplicación.

4.4.4.2. Técnicas de monitorización software

Las técnicas para la monitorización de software pese a ser muy específicas de cada diseño, se pueden establecer una serie de reglas o recomendaciones para guiar la monitorización de software.

Estas recomendaciones se pueden separar en varios tipos, como recomendaciones para la monitorización del software encargado de gestionar recursos hardware:

- Gestión adecuada de las interrupciones del sistema.
- Cumplimiento de los requisitos de tiempos de ejecución.
- Correcto tratamiento de los transitorios en secuencia de arranque del sistema.
- Gestión adecuada de los transitorios en las señales de interfaces, por ejemplo buses.
- Control de los tiempos de accesos de lectura, escritura y asignación de espacio en memorias.
- Gestión de violaciones entre las diferentes particiones de software.
- El comportamiento de los lazos de control.
- Comprobación de la validez y la compatibilidad de software actualizable por el usuario.

Además se establecen recomendaciones para la monitorización interna del software:

- Inicializaciones adecuadas de las variables del sistema.

- Control de los rangos y resolución de las variables.
- Gestión de la corrupción de datos.
- Control de las secuencia de eventos y operaciones.
- Precisión y exactitud de los valores productos de algoritmos.
- Gestión adecuada de bucles.

4.4.5. Técnicas BITE para sistemas modulares integrados

En el desarrollo de sistemas modulares integrados [24], [25], en la implementación de capacidades de BITE se utilizan los siguientes métodos de monitorización para los distintos niveles del computador IMA.

El nivel de criticidad definido para las aplicaciones debe de ser el mismo que para el resto del sistema integrado. Por lo tanto los requisitos para detección de condiciones de fallo deben de ser los mismos para la aplicación software que para el resto del sistema.

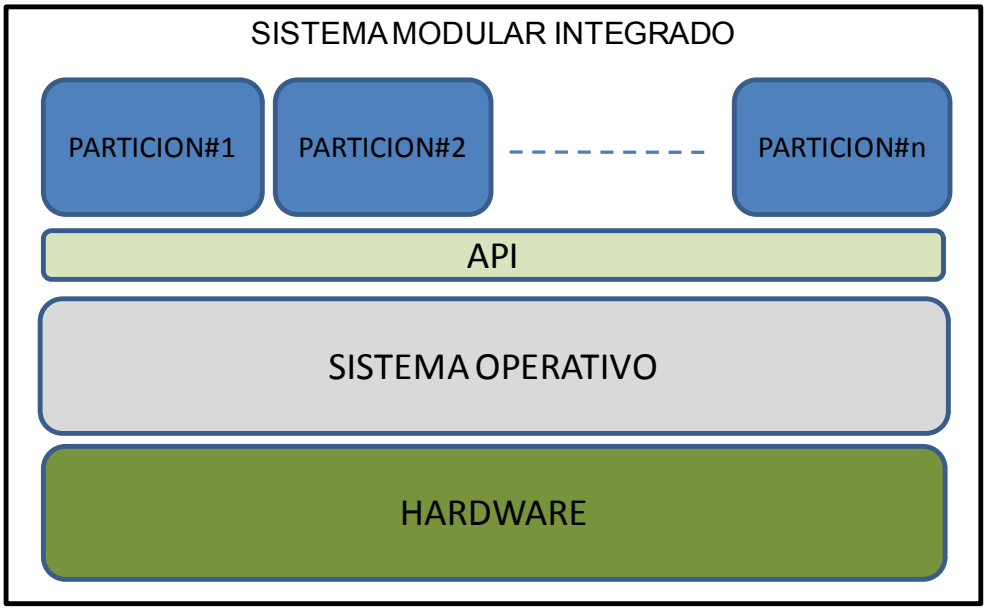


Figura 6. Esquema de un Sistema Modular Integrado

4.4.5.1. Monitorización de errores a nivel partición

A nivel partición, ver Figura 6, el Sistema Modular Integrado debe monitorizar y gestionar los siguientes tipos de errores:

- Configuración durante la inicialización de la partición.
- Instalación en la partición.
- Gestión de procesos.
- Operación de procesos.

Cuando un fallo es detectado en una partición, en función de la criticidad del fallo, la partición puede reconfigurarse, pasar a un modo degradado, incluso reiniciarse o detenerse de acuerdo a los requisitos especificados para la gestión de dicho fallo.

El objetivo de la capacidad de reconfiguración de la partición, es que un fallo detectado a este nivel, no tenga consecuencias en la plataforma donde está embarcado el sistema.

4.4.5.2. Monitorización de errores a nivel de procesos software

La monitorización de los procesos software de las diferentes aplicaciones embarcadas en el Sistema Modular Integrado estarán diseñados de acuerdo a los requisitos del sistema, y siguiendo las técnicas aplicables a la monitorización de software, ver apartado 4.4.4.

4.4.5.3. Monitorización de la capa API

La capa Interfaz del Programa Aplicación (API), ver Figura 6, es la interfaz entre la aplicación y el hardware del Sistema Modular Integrado. La capa API tiene como objetivo garantizar que los recursos del sistema modular están disponibles cuando sean requeridos por las aplicaciones. Por lo tanto, deben estar disponibles durante el tiempo de operación y deben estar monitorizados.

Para ello, la capa API es la encargada de gestionar las particiones, procesos, tiempos, memoria, comunicación entre particiones, y monitorización de fallos.

Cada vez que una aplicación realiza una llamada al API, este comprueba la disponibilidad de los recursos solicitados. En caso de encontrar algún fallo, el API debe ejecutar la correspondiente reconfiguración y enviar el código de fallo correspondiente, de acuerdo a la especificación de BITE del Sistema Modular Integrado, además de almacenar internamente la información relativa al fallo detectado.

4.4.6. Generación y análisis de patrones de test

Uno de los métodos más efectivos para la comprobación de fallos en un circuito o en un sistema, es el método de generación de patrones de test, gracias a su alta cobertura de fallos [31]. Este método está basado en el diseño para test (DTF), ver apartado 4.4.2.

Aunque originalmente este método de generación y análisis de patrones de test también conocido como BIST, se utilizaba únicamente en circuitos digitales complejos, actualmente se utiliza a nivel circuito o sistema. Por este motivo, en este TFM se tratará la implementación del BIST a nivel dispositivo.

Anteriormente a la incorporación de BIST en los circuitos o sistemas, se recurría a la utilización de un sistema externo, que mediante la inyección de patrones a través de las interfaces del sistema o circuito, realizaba los test necesarios para la comprobación del sistema o circuito.

La técnica BIST proporciona al dispositivo la capacidad de generar patrones de test, aplicarlos directamente al dispositivo y evaluar si los resultados de los test son los esperados. La Figura 7, muestra el esquema de arquitectura de un BIST.

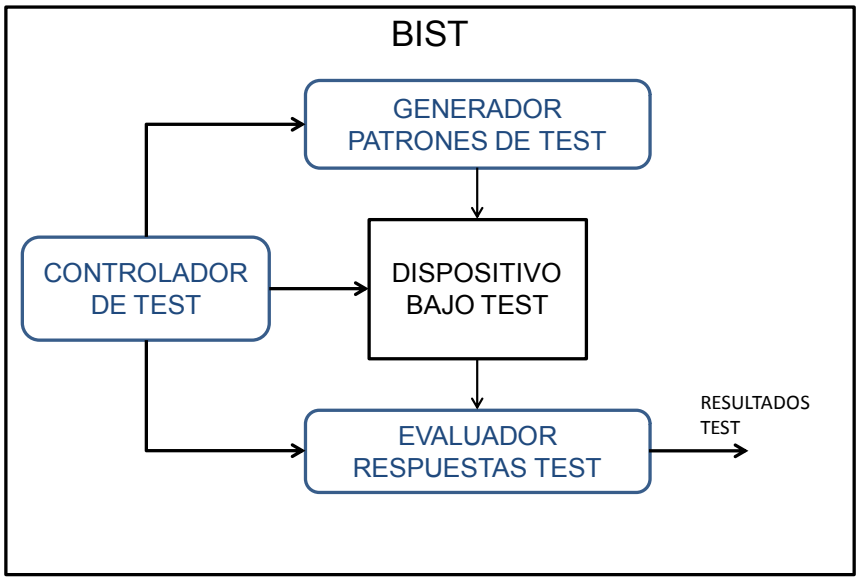


Figura 7. Esquema de un sistema BIST.

La implementación de una arquitectura BIST en un dispositivo, proporciona la capacidad de realizar sobre el dispositivo, test de tipo funcional y test de diagnóstico.

Como muestra la Figura 7, la implementación de la técnica BIST sobre un dispositivo, requiere de un hardware adicional. Este hardware adicional se puede dividir en tres bloques:

- Controlador de test.
- Generador de patrones de test.
- Analizador de las respuestas.

A la hora de plantear el diseño y desarrollo de un dispositivo con capacidad BIST, es necesario tener en consideración los siguientes datos:

- Cobertura de fallos.
- Tamaño del conjunto de test.
- Hardware adicional.
- Impacto en el funcionamiento del dispositivo.

En función de la cobertura de fallos requerida para el dispositivo bajo test, serán necesarios un cierto número de patrones de test. De acuerdo a los test necesarios, se define el hardware adicional para el BIST y por consiguiente el impacto en el funcionamiento del dispositivo.

Con el fin de minimizar los recursos necesarios para la implementación del BIST, el evaluador de respuestas a los patrones de test, compacta los resultados de los test y evalúa el resultado compactado con el resultado esperado. De este modo se reduce los recursos necesarios para esta función del BIST.

La capacidad de BIST sobre un dispositivo funciona del siguiente modo. Durante el funcionamiento del dispositivo en modo de test, el controlador de test configura el dispositivo bajo test en un modo determinado. A continuación, el generador de patrones de test, gestionado por el controlador, aplica una secuencia de patrones de test al dispositivo bajo test. De este modo el evaluador de respuestas recibe y evalúa el comportamiento del dispositivo bajo test a cada patrón de test, identificando si el dispositivo bajo test, presenta alguna condición de fallo.

En caso de que el dispositivo bajo test presente alguna condición de fallo, el BIST, generará el mensaje de fallo asociado.

Los principales beneficios que proporcionan la implementación de técnicas BIST en un dispositivo son los siguientes:

- Desde el punto de vista del fabricante de circuitos digitales o de sistemas, la incorporación de capacidad BIST puede ayudar a simplificar el proceso de caracterización del dispositivo, proporcionando una mayor visibilidad de su operación. Además, dado que el propio circuito dispone de capacidad de autocomprobación, permite la reducción del tiempo los test de fabricación.

- La comprobabilidad del equipo se realiza de forma rápida y eficiente gracias a un proceso jerárquico de comprobación.

- El BIST permite comprobaciones durante la operación del sistema en tiempo real y en modo mantenimiento, mientras el sistema no está en operación.

- No es necesario la utilización de equipos de test adicionales.

Por el contrario, la integración de un BIST en un dispositivo, también lleva consigo una serie de desventajas como:

- Necesidad de incorporar circuitería e interfaces adicionales, añadiendo complejidad al dispositivo.

- La operación del dispositivo debe modificarse para ser capaz de gestionar el BIST.

- Por último todo esto lleva consigo un mayor coste del diseño, desarrollo y fabricación del dispositivo.

5 Entorno de Mantenimiento Centralizado en plataformas de sistemas modulares

En los capítulos anteriores se ha desarrollado el concepto de BITE como solución para el mantenimiento de en un sistema. En este capítulo se va ampliar la utilización de la capacidad de BITE para todos los sistemas de una plataforma multisistema, de forma estandarizada, y haciendo que todos los módulos BITE de los diferentes sistemas estén conectados con un sistema central, estableciendo un Entorno de Mantenimiento Centralizado.

5.1. Descripción y necesidades de la industria

En la actualidad, entre los objetivos principales de cualquier industria se encuentran el maximizar el beneficio y el minimizar los costes. Para conseguir ambos objetivos, dos pasos fundamentales son incrementar del rendimiento y la eficiencia.

En la industria para incrementar del rendimiento y la eficiencia es importante mantener sus sistemas en operación el mayor tiempo posible, minimizando los tiempos de mantenimiento, lo cual es cada vez más complicado, dada la elevada complejidad de las diferentes plataformas, junto a las diversas y complejas localizaciones de los sistemas a lo largo de la plataforma.

Desde el punto de vista del mantenimiento de sistemas, aprovechando las ventajas de la constante evolución de la tecnología, con el fin de alcanzar las demandas de la industria, los diseñadores y fabricantes de sistemas tienen el reto constante de conseguir reducir y facilitar las tareas de mantenimiento de los sistemas, pese al aumento de la complejidad de dichos sistemas.

Para ello, el desarrollo de las capacidades de comprobación y diagnóstico junto con la comunicación entre sistemas, está permitiendo la implementación del mantenimiento centralizado en las plataformas multisistema. La base del mantenimiento centralizado, está basada en la integración de las capacidades de BITE de todos los sistema de una plataforma, con el fin de realizar una gestión de mantenimiento centralizada desde un único sistema.

Este único sistema que centraliza el mantenimiento de la plataforma, está basado en gestionar de forma continua la información de BITE de cada uno de

los sistemas de la plataforma. Para ello, cada sistema de forma periódica ejecuta una serie de test que verifican que la unidad está funcionando dentro de sus márgenes operacionales. El resultado de esos test, son enviados al Sistema de Mantenimiento Centralizado, permitiendo la detección inmediata de un fallo.

La criticidad del entorno de mantenimiento es proporcional a la criticidad de la plataforma en la que está instalado este sistema.

El Entorno de Mantenimiento Centralizado (EMC) en una plataforma multisistema tiene como objetivos principales la gestión y el control del mantenimiento de forma integral para todos los sistemas que forman parte de dicha plataforma.

Forman parte del EMC todas las funciones, módulos, equipos, o sistemas que están relacionados con el mantenimiento dentro en la plataforma multisistema.

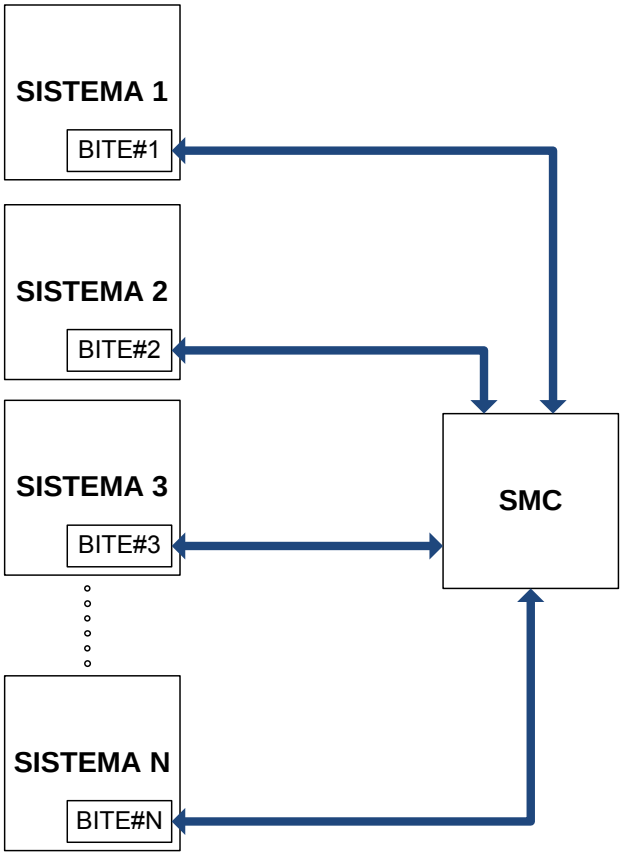


Figura 8. Entorno de Mantenimiento Centralizado.

La base del EMC reside en la gestión centralizada de las funciones Built-In-Test Equipment (BITE) de cada sistema, por medio de un Sistema de Mantenimiento Centralizado (SMC) [7], [8], como indica la Figura 8.

El EMC tiene como función principal recopilar la información de BITE de todos los sistemas de la plataforma y proporcionar al personal de mantenimiento la información necesaria para realizar las tareas de mantenimiento de forma rápida y precisa, evitando ambigüedades en la identificación del equipo fallados con el fin de restablecer, las condiciones óptimas de operación en todos los sistemas de la plataforma.

Es importante remarcar que el EMC está exclusivamente orientado al mantenimiento, por lo tanto otras funciones disponibles durante la operación de la plataforma, como gestión de avisos y alertas, reconfiguraciones por degradación de sistemas, etc., están fuera del perímetro del EMC.

Complementado los ejemplos introducidos en el capítulo 3, Estado del Arte, los EMC se están comenzando a introducir en los nuevos desarrollos de plataformas multisistemas de diferentes industrias como: aeronaves, trenes, automóviles, barcos, edificios inteligentes, plantas de fabricación y plantas de generación de energía.

5.2. Arquitectura del Entorno de Mantenimiento Centralizado

En la definición del diseño de la arquitectura de un EMC está establecida en dos niveles:

- Arquitectura básica: Este nivel de arquitectura incluye los equipos mínimos para establecer el EMC, ver Figura 9.
- Arquitectura extendida: Este segundo nivel de arquitectura incluye otros equipos y/o funciones periféricos que incrementan las capacidades del EMC, ver Figura 10.

Si la criticidad de los sistemas que forman parte de la plataforma lo requiere, el SMC podría estar duplicado.

5.2.1. Arquitectura básica del EMC

Teniendo en cuenta la función principal del EMC, la arquitectura básica de un EMC debe de estar compuesta por los siguientes componentes:

- Partición BITE de cada sistema de la plataforma multisistema.
- Interfaces entre el módulo BITE de cada sistema y el SMC.
- SMC, formado por los siguientes componentes o módulos:
  - Módulo de entrada salida.
  - Módulo procesador.
  - Módulo de control y visualización.
  - Base de datos.

La Figura 9, muestra de forma gráfica la arquitectura básica del EMC.

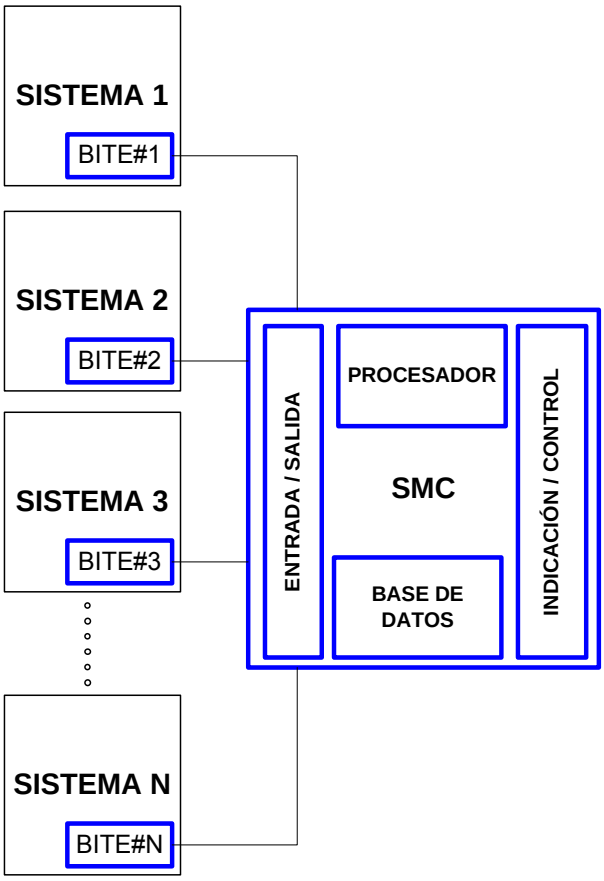


Figura 9. Arquitectura básica de un EMC.

5.2.2. Arquitectura extendida del EMC

Dependiendo de los requisitos del EMC y de la plataforma en la que esté instalado, este puede disponer de capacidades adicionales, como:

5.2.2.1. Comunicación con el exterior de la plataforma

Esta capacidad se utiliza en algunas plataformas que forman parte de un metasistema. (Ver apartado 5.5). Dependiendo de la plataforma, la comunicación con el exterior puede ser vía radio, teléfono, enlace de datos, satélite o TCP/IP.

5.2.2.2. Utilización de dispositivos portátiles

Cada día es más común la utilización de dispositivos portátiles como tabletas o equipos específicos para el mantenimiento de plataformas.

Para posibilitar el envío de información desde el EMC a este tipo de dispositivos, es necesario que el SMC disponga de una interfaz para este tipo de conexiones. Aunque las interfaces típicas son USB, WIFI, pueden disponer de interfaces particulares de la industria en la que esté implementado el EMC.

5.2.2.3. Almacenamiento de datos en dispositivos externos

Esta capacidad permite la exportación de los informes de mantenimiento y otros datos del SMC a dispositivos externos de almacenamiento de datos. Para ello el SMC, al igual que para el uso de dispositivos portátiles, dispondrá de una interfaz dedicada.

5.2.2.4. Control de acceso al EMC

Acorde con la criticidad de la plataforma, el acceso al SMC puede estar restringido. La gestión del control de acceso al SMC, se puede realizar de diferentes modos:

- Acceso con un Usuario y contraseña:

  Este tipo de control de acceso se gestiona directamente a través de la interfaz de visualización y control de usuario.

- Acceso con tarjeta de identificación, huella dactilar, etc.:

  La gestión del acceso a través de estos tipos de identificaciones, requieren que el SMC disponga de una interfaz con un dispositivo lector de tarjetas, huellas o para otro tipo de identificación.

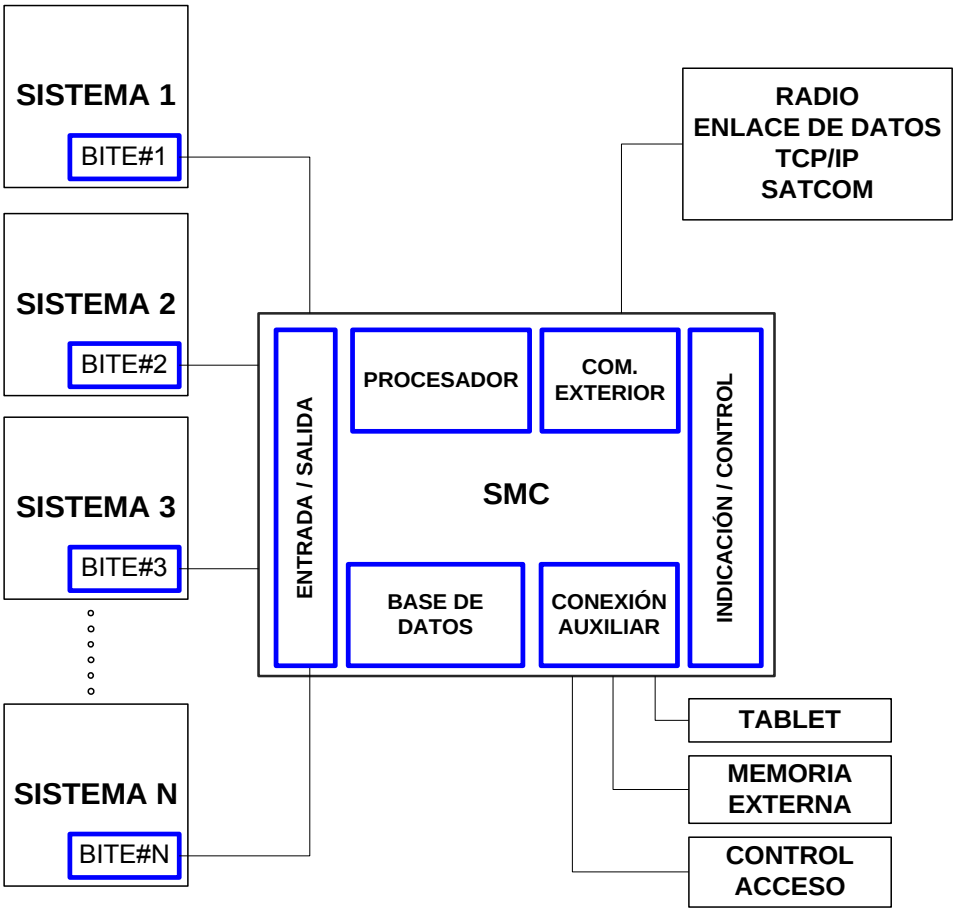


Figura 10. Arquitectura extendida de un EMC.

5.2.3. Sistema de Mantenimiento Centralizado

El componente principal del EMC es el SMC. Dentro de la plataforma multisistema, el SMC es un sistema más, el cual está integrado con todos los

demás sistemas de la plataforma multisistema a través de las interfaces de comunicación, como se muestra en las Figura 8, Figura 9 y Figura 10.

El SMC está compuesto por una parte software y una parte hardware.

La parte software esta empotrada dentro de este hardware y es la encargada de ejecutar las siguientes funciones del SMC, se encuentran desarrolladas en el apartado 5.3:

- Gestor de los Datos de Mantenimiento.
- Acceso a los Datos de Mantenimiento.
- Gestor del Aislamiento de Fallos.
- Acceso a los Test de Diagnóstico (iBITE).
- Generación de Informes de Mantenimiento.

Adicionalmente el SMC puede tener otros softwares encargados con otras funciones como por ejemplo definir la configuración de la plataforma multisistema.

Con el fin de implementar actualizaciones o cambios de configuración en la plataforma, el SMC debe permitir la actualización del software en el propio equipo [3].

La parte hardware está definida de acuerdo a la arquitectura mostrada en la Figura 11 y cuyos componentes se describen a continuación.

A nivel hardware la Figura 11, muestra los principales componentes de un SMC:

- Módulo de indicación y control.
- Módulo de procesado.
- Base de datos (Memoria no volátil).
- Módulo de entrada / salida.

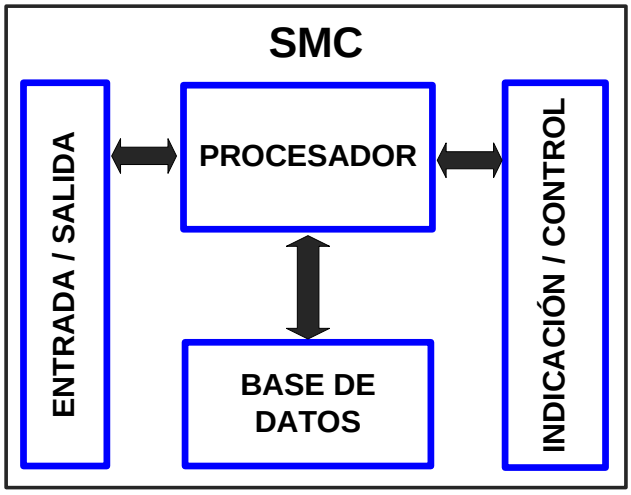


Figura 11. Arquitectura de un Sistema de Mantenimiento Centralizado.

5.2.3.1. Módulo de indicación y control del SMC

Proporciona la interfaz visual y de control entre el sistema SMC y el personal de mantenimiento. Esta interfaz permite el acceso a la información de mantenimiento recibida de cada uno de los sistemas de la plataforma y procesada por el SMC.

Este módulo estará formado por una pantalla para la presentación de la información y un control. El control puede implementarse a través de un dispositivo tipo ratón, mediante pulsadores alrededor de la pantalla o por medio de una pantalla táctil. La interfaz con el usuario debe ser sencilla, clara y auto explicativa, con el fin de evitar que el usuario pueda equivocarse. En el apartado 5.4, se describen los menús de la interfaz, a través de los cuales se da acceso a las distintas funciones del SMC.

5.2.3.2. Módulo de entrada / salida del SMC

En un EMC, el SMC tiene interfaces con todos los sistemas de la plataforma multisistema. Estas interfaces son bidireccionales con el fin de que cada sistema de la plataforma pueda reportar periódicamente su información de BITE y recibir del SMC los diferentes comandos necesarios para ejecutar los test de diagnóstico y aislamiento de fallos.

El protocolo de comunicación utilizado en el EMC entre los sistemas y el SMC debe de estar estandarizado para todos los sistemas de la plataforma.

Dependiendo de la plataforma, el módulo entrada / salida puede tener interfaces con el resto de sistemas a través de buses tipo: TCP/IP, A429, AFDX, 1553, CAN Bus, etc.

5.2.3.3. Mensaje de fallo

La estructura del mensaje de fallo debe de ser común para todos los sistemas de la plataforma. En la siguiente tabla se muestra un el contenido típico de un mensaje de fallo.

Mensaje
Sistema
Identificador del sistema
Número de parte
Número de Serie
Código de Fallo
Alerta asociada
Clase del fallo
Unidad Acusada 1
Unidad Acusada 2
Unidad Acusada 3
Conector acusado 1
Contacto acusado 1
Conector acusado 2
Contacto acusado 2
Conector acusado 3
Contacto acusado 3
Fecha y hora

Tabla II. Contenido de un mensaje de fallo para el SMC.

5.2.3.4. Base de datos del SMC

Los datos recibidos por el SMC, de cada sistema de la plataforma, son procesados y almacenados en una base de datos la cual almacena su contenido en una memoria de tipo no volátil. El tamaño de esta memoria estará dimensionado de acuerdo a la máxima cantidad de datos que requiera almacenar el SMC.

Para facilitar la gestión de los fallos reportados por cada sistema, es aconsejable realizar el almacenamiento en la base de datos de forma estructurada. La Figura 12 muestra una estructura recomendada para la base de datos del SMC.

En primer lugar se ha dividió la base de datos de almacenamiento en tres, partes, la primera, para almacenar los datos de la sesión actual, la segunda para almacenar el histórico de los datos del SMC y por ultimo una zona reservada para el fabricante o diseñador, con el fin de almacenar información relativa al sistema.

Es recomendable almacenar por separado la información reportada por cada uno de los sistemas de forma automática, a través de los test de inicio o de los test en modo continuo, de la información reportada por cada uno de los sistemas en modo interactivo.

Adicionalmente, la información de cada sistema se almacena por separado en cada una de las zonas de la base de datos.

Tanto en la zona de almacenamiento de la sesión actual, como en la zona de almacenamiento histórico, hay un espacio reservado para el almacenamiento los parámetros característicos de la plataforma multisistema, que deban ser grabados de acuerdo con las reglas definidas.

Con la base de datos estructurada de este modo, durante la sesión en curso de la plataforma, el SCM registrará los datos recibidos, en la zona de sesión actual y cuando finalice esta sesión, el sistema volcara estos datos a la zona de almacenamiento histórico.

De este modo, se gestiona el acceso a la información almacenada de forma eficaz.

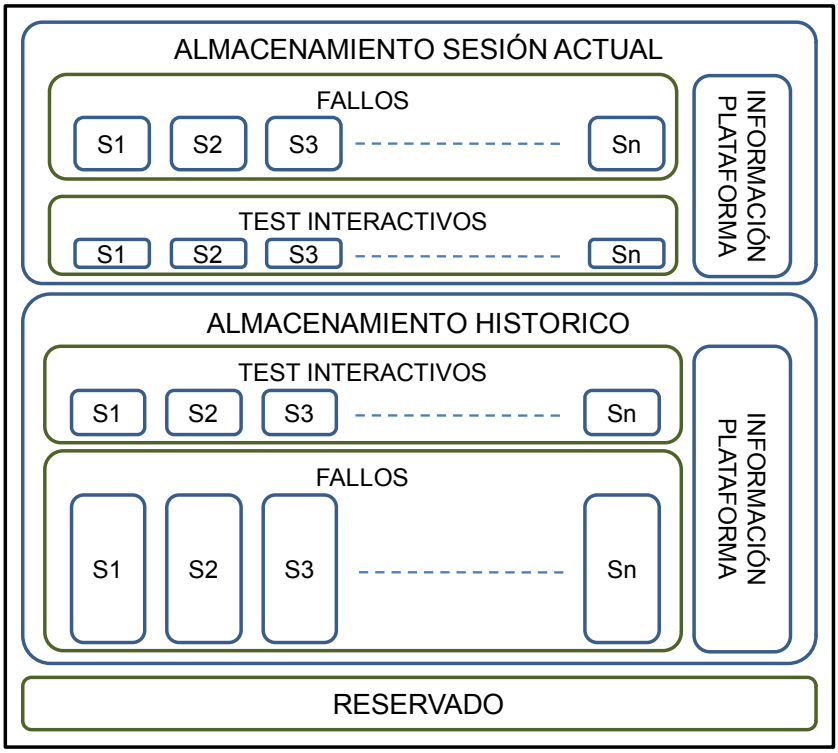


Figura 12. Estructura de la base de datos del SMC.

5.2.3.5. Procesador del SMC

El SMC dispondrá de un módulo procesador sobre el que se ejecuta el software que permite el funcionamiento de las diferentes funciones del SMC.

Además, este módulo procesador, se es el encargado de gestionar y procesar la información recibida y enviada por el módulo entrada/salida, junto con la gestión del módulo de indicación y control.

5.2.4. Módulos BITE de cada sistema

Cada sistema de la plataforma, dispone de un módulo BITE. Las principales funciones de este módulo BITE de cada sistema son:

- Monitorizar el estado del sistema y en caso de detectar alguna discrepancia

- Procesar el fallo detectado y compilar la información relativa a este fallo para reportarla al SMC junto con los datos del sistema, de acuerdo a la estructura mostrada en la Tabla II.
- Enviar la información al SMC a través de la interfaz definida.
- El módulo BITE de cada sistema también debe disponer de la capacidad de recibir comandos desde el SMC con el fin de ejecutar los test interactivos.

Aunque cada sistema del EMC, tenga diferentes funciones, y este diseñado bajo distintas filosofías, el diseño del módulo BITE debe estar especificado de acuerdo a un estándar común para todos los sistemas de la plataforma.

5.3. Funciones del Sistema de Mantenimiento Centralizado (SMC)

En este apartado se describen las principales funciones que puede tener un SMC. La Figura 13 muestra esquemáticamente las diferentes funciones de un Sistema de Mantenimiento Centralizado.

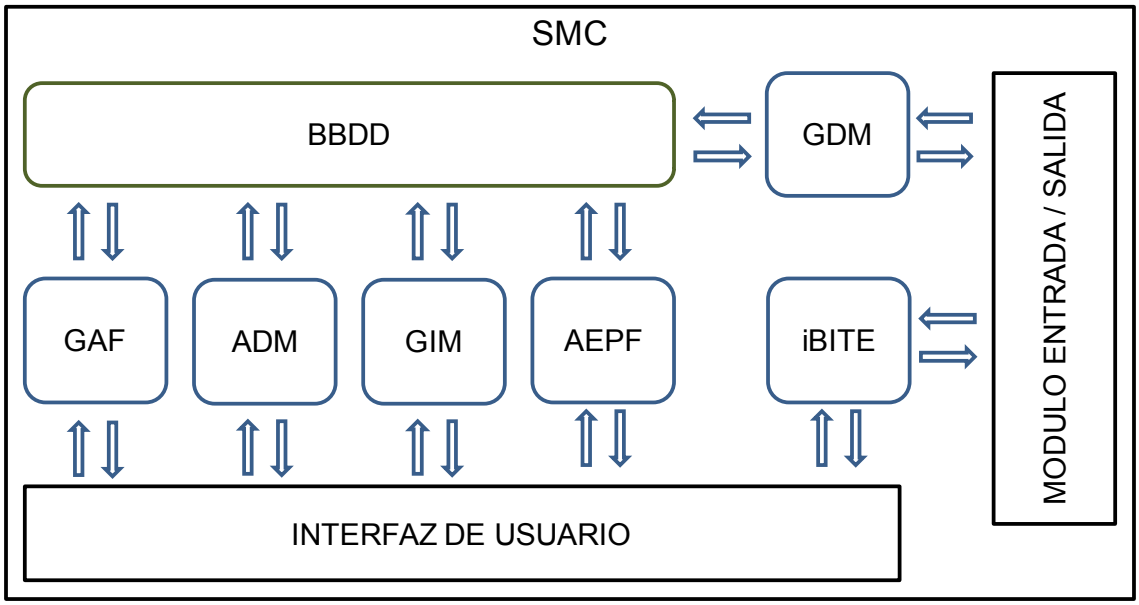


Figura 13. Funciones de un Entorno de Mantenimiento Centralizado.

5.3.1. Gestor de los Datos de Mantenimiento (GDM)

La función Gestor de los Datos de Mantenimiento (GDM) para cumplir con su objetivo, se encarga de recopilar los datos de mantenimiento proporcionados por cada uno de los sistemas de la plataforma a través de sus interfaces, procesar la información recibida y almacenarla en la base de datos correspondiente.

Esta función GDM además proporciona otras capacidades como la monitorización y gestión de otros parámetros significativos de la plataforma, los cuales pueden resultar importantes para el mantenimiento como pueden ser las alertas de fallos, o degradaciones junto con sus respectivas marcas de tiempo.

Los datos de entrada de la función GDM son los siguientes:

- Información de los fallos: La función BITE de cada sistema de la plataforma, envía la información de los fallos detectados al SMC.
- Alertas de fallo: El SMC monitoriza las diferentes posibles alertas de la plataforma gestionadas por un Sistema de Alertas, este sistema es externo al SMC. La función GDM correla las Alertas de fallos mostradas, con los códigos de fallos proporcionados por los sistemas.
- Parámetros de operación: Son el conjunto de parámetros característicos de la plataforma, los cuales permiten definir las principales fases o modos de operación de la plataforma en un cierto momento.
- Parámetros característicos de la plataforma: Son el conjunto de parámetros que permiten diferentes condiciones de la plataforma en el momento de un fallo. Como ejemplo de parámetros característicos típicos en una plataforma, pueden ser los parámetros ambientales como temperatura, humedad, presión.

La función GDM tras la gestión de los datos de entrada debe proporcionar los datos procesados de acuerdo a un formato predefinido con el fin de permitir su almacenamiento en la base de datos del sistema:

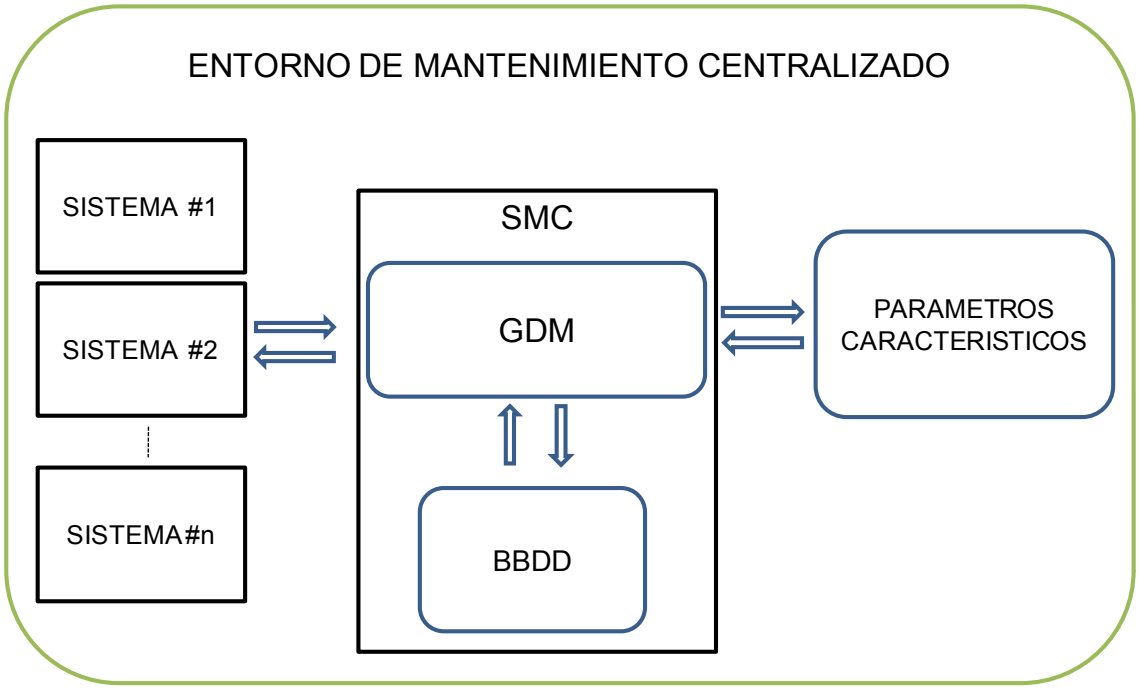


Figura 14. Esquema de la función Gestor de los Datos de Mantenimiento.

5.3.2. Gestor del Aislamiento de Fallos (GAF)

La función Gestor del Aislamiento de Fallos (GAF) es la encargada de leer y procesar la información almacenada en la base de datos del SMC.

El procesado de la información que realiza la función GAF, consiste en una correlación de los diferentes datos registrados por el sistema. Para ello y de acuerdo a una serie de algoritmos especificados, se comprueban si hay relación temporal entre los mensajes de fallos reportados por los diferentes sistemas, las alertas producidas en la plataforma y con posibles eventos de acuerdo al análisis de los parámetros de operación y parámetros característicos de la plataforma.

Un ejemplo característico de correlación de fallos relativos a alimentación de los sistemas. Cuando varios sistemas en un mismo instante de tiempo reportan códigos de fallos que acusan a los módulos de alimentación de los respectivos sistemas, es muy probable que el origen de los fallos sea una incidencia en la alimentación global de la plataforma y no en los módulos de alimentación de los diferentes sistemas.

Con los datos procesados, el SMC genera informes de mantenimiento con la información concreta de las tareas requeridas, para volver a poner la plataforma completamente operativa.

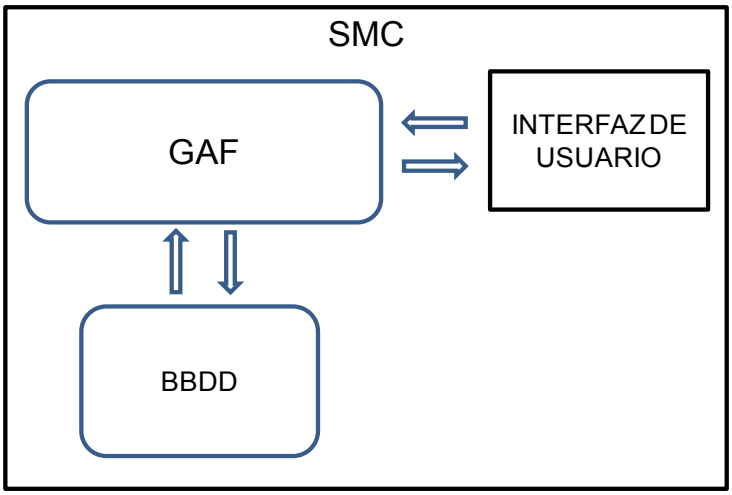


Figura 15. Esquema de la función Gestor del Aislamiento de Fallos.

5.3.3. Acceso a los Datos de Mantenimiento (ADM)

La función de Acceso a los Datos de Mantenimiento (ADM) es la encargada de proporcionar al operador de mantenimiento, la información disponible en el SMC a través de la interfaz de usuario. A través de las diferentes opciones en los menús, el operador de mantenimiento, puede acceder a la información almacenada en la base de datos.

La Figura 16 presenta de forma esquemática las interfaces de la función ADM. Esta función permite presentar los datos en distintos formatos de acuerdo a los equipos periféricos de que disponga el SMC. Por norma general, el usuario puede visualizar los datos de mantenimiento en la pantalla de la interfaz de usuario, enviarlos a la impresora del sistema, descargarlo en formato electrónico a través de un dispositivo portátil de almacenamiento de datos, o enviarlos a otro sistema.

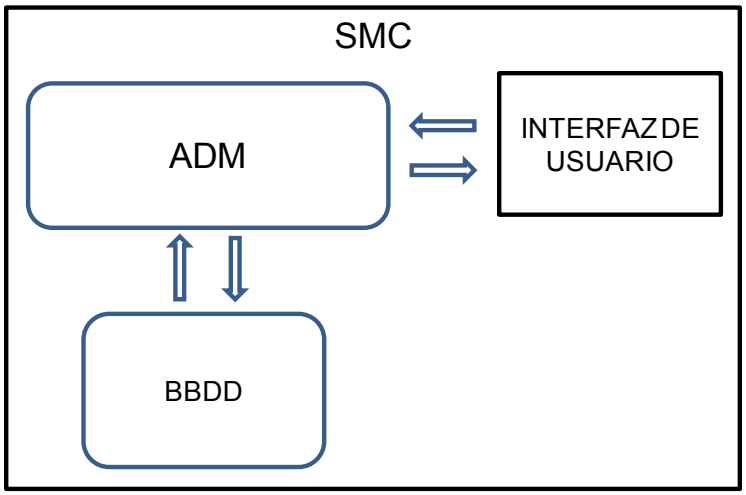


Figura 16. Esquema de la función Acceso a los Datos de Mantenimiento.

5.3.4. Acceso a los Test de Diagnóstico (iBITE)

La función de Acceso a los Test de Diagnóstico (iBITE) gestiona el acceso a los test interactivos de cada sistema. La función iBITE, puede acceder a la interfaz con todos los sistemas de la plataforma, y a la interfaz de usuario como se indica en la Figura 17. Esta función envía un comando al sistema seleccionado a través de la interfaz de usuario, solicitando el test interactivo. Cuando el sistema seleccionado finaliza el test, envía el resultado al SMC, y presenta los resultados al operador.

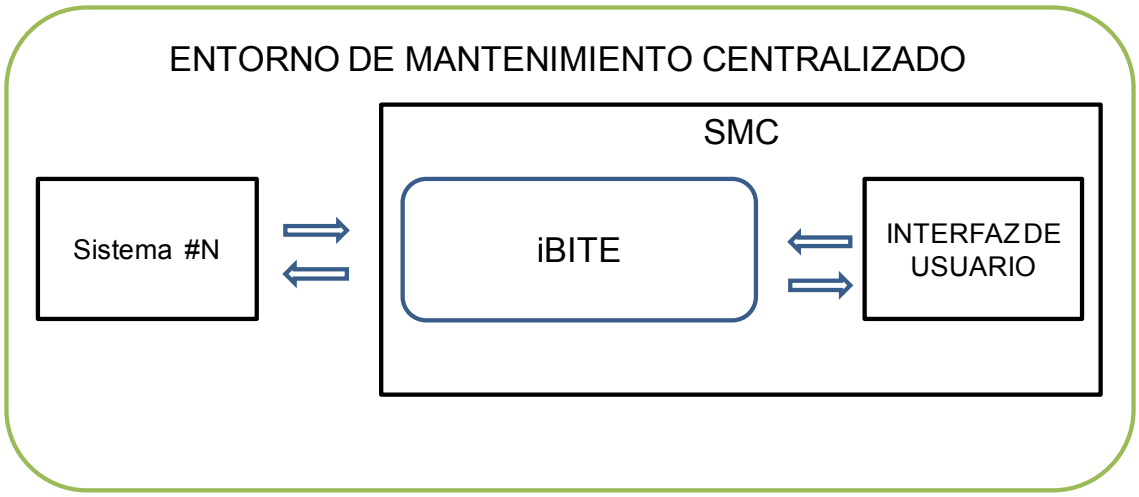


Figura 17. Esquema de la función Acceso a los Test de Diagnóstico (iBITE).

El acceso a los test de la función iBITE está restringido para cuando la plataforma está en modo de mantenimiento.

5.3.5. Generación de Informes de Mantenimiento (GIM)

La función Generación de Informes de Mantenimiento (GIM) es la encargada de proporcionar al operador de mantenimiento, la capacidad de generar informes de mantenimiento de la plataforma multisistema.

Para ello la función GIM, de acuerdo a los requisitos de los datos solicitados, accede a la información almacenada en la base de datos del SMC, para procesar y presentar los datos requeridos en el informe de mantenimiento. De este modo, el operador de mantenimiento, siguiendo el informe de mantenimiento generado, identifica que tareas de mantenimiento son necesarias para restaurar la operación adecuada de todos los sistemas de la plataforma multisistema.

La información que puede presentar el SMC en un informe será acorde con la información que proporcionen las funciones del SMC.

Dependiendo de las capacidades del SMC, el contenido de los informes de mantenimiento puede estar predefinido, o puede permitir al operador de mantenimiento seleccionar la información que desea incluir en dichos informes. El contenido de los informes de mantenimiento podría ser configurado y cargado al SMC a través de un *User Modifiable Software* (UMS) [3].

Las opciones de presentación de los informes, al igual que la función ADM, dependen de las interfaces periféricas del SMC.

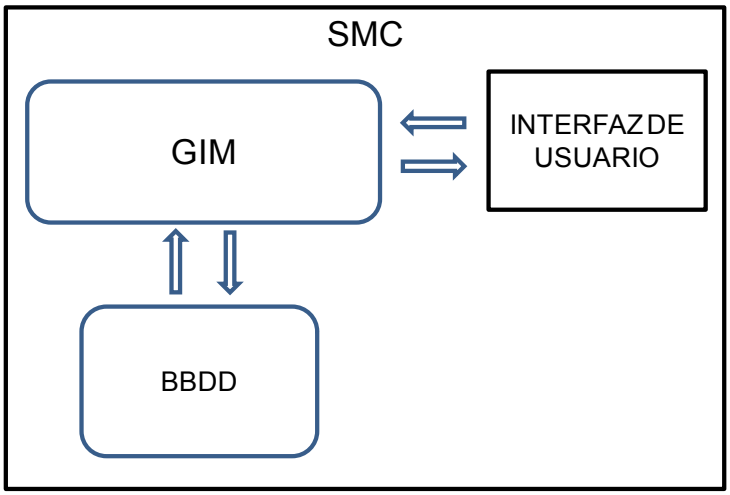


Figura 18. Esquema de la función Generador de Informes de Mantenimiento.

5.3.6. Análisis Estadístico para la Predicción de Fallos (AEPF)

La función Análisis Estadístico para la Predicción de Fallos (AEPF), proporciona al operador de mantenimiento con un análisis estadístico de los datos almacenados por el SMC con el fin de predecir el tiempo de vida de un equipo.

La función AEPF como muestra la Figura 19 tiene interfaces la base de datos del SMC y con la interfaz de usuario.

Cuando el operador de mantenimiento a través de la interfaz de usuario selecciona un equipo a analizar de un sistema determinado, la función AEPF en base a los códigos de fallo almacenados en el SMC, que acusan al equipo seleccionado como origen de un fallo, calcula la función de probabilidad con el fin de predecir el tiempo de vida estimado para el equipo analizado.

Este TFM propone la distribución de probabilidad de Weibull (ver apartado 5.6), como herramienta estadística para la función AEPF.

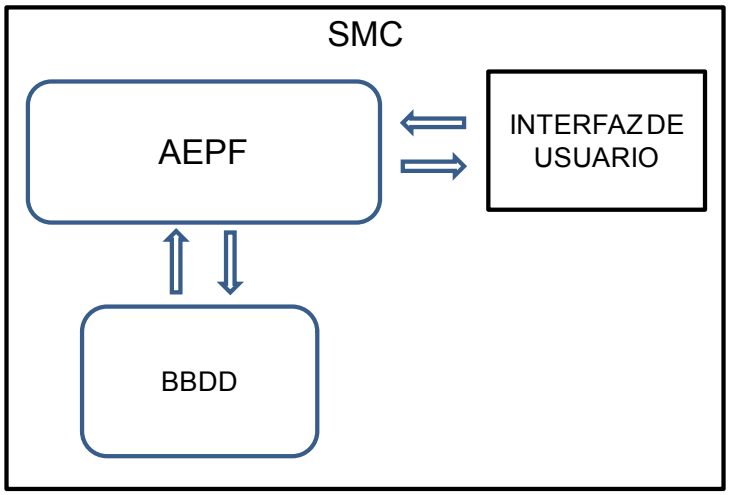


Figura 19. Esquema de la función Análisis Estadístico para la Predicción de Fallos.

5.4. Control y Operación del EMC

El EMC dispone de dos modos principales de funcionamiento, durante la operación de la plataforma multisistema, el EMC trabaja en modo Monitorización y cuando la plataforma detiene su operación, el EMC permite activar el modo Mantenimiento. El operador de mantenimiento tendrá acceso a las diferentes funciones del EMC a través de una interfaz dedicada del SMC, similar a la mostrada en la Figura 20.

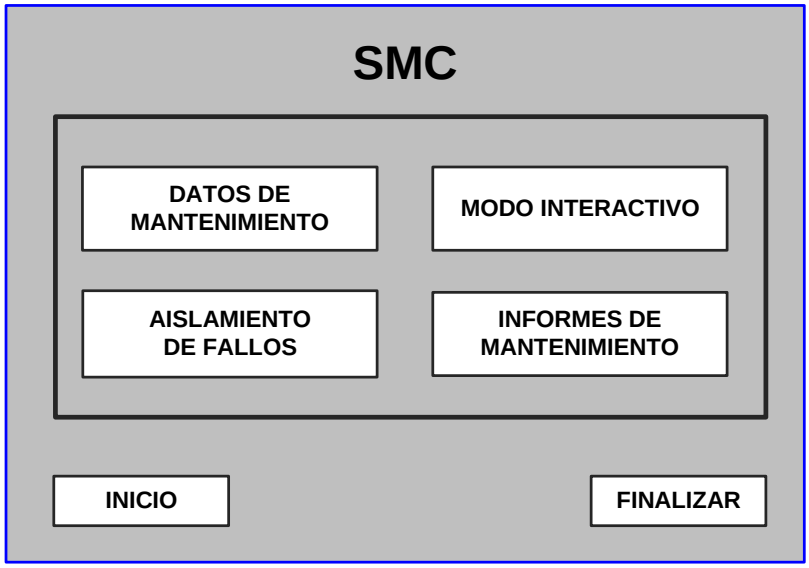


Figura 20. Interfaz de usuario del SMC.

5.4.1. Monitorización

El EMC de forma continua, recibe y procesa la información de mantenimiento que envía el módulo BITE de cada sistema de la plataforma.

Durante la monitorización, el BITE de cada sistema puede enviar información relativa al Test de Inicio, tras el encendido del sistema y a continuación los datos de BITE del Modo Continuo, ver apartado 4.4.1.3.

- Test de inicio:

  Cuando un sistema se inicializa, lo primero que hace es realizar un test de inicio del propio sistema, reportando cualquier fallo que pueda afectar a la seguridad del sistema. Estos fallos se reportan del mismo modo que los fallos que ocurran durante el resto de la operación normal del sistema.

- Modo Continuo:

  Está basado en la monitorización continua de cada unidad del sistema. En cada unidad del sistema, el BITE ejecuta periódicamente una serie de test que verifican que la unidad está funcionando dentro de sus márgenes operacionales. Si durante la operación el BITE de un sistema detecta algún fallo, enviara la información relativa a dicho fallo al EMC. En función de la criticidad del fallo otros sistemas de gestión de alarmas pueden llevar a cabo distintas acciones como reconfiguración del sistema, operación con limitaciones en modo degradado, etc. antes de detener el funcionamiento de la plataforma para la sustitución de la unidad fallada.

5.4.2. Mantenimiento

Una vez que la plataforma finaliza la operación, se puede iniciar el modo mantenimiento del EMC. En este modo, el operador de mantenimiento puede acceder a las distintas funciones del EMC analizadas en el apartado 5.3 y en relación a los sistemas lanzar los test de Modo Interactivo de los diferentes sistemas de la plataforma ver apartado 4.4.1.3.

El Modo Interactivo sólo puede ser activado si la plataforma multisistema no se encuentra en fase de operación. (ej.: en un avión el modo interactivo solo está accesible si el avión está en tierra y con motores apagados.)

5.5. Meta sistema

La arquitectura desarrollada para un EMC, permite escalarla a un nivel superior y desarrollar el concepto de Meta-sistema para varios EMC.

Al igual que el desarrollo de un EMC está basado en el concepto de compartir información entre todos los sistemas de una plataforma, para el desarrollo de un Meta-Sistema de EMC para múltiples plataformas multisistema se debe aplicar el mismo concepto.

Como se muestra en la Figura 21, el Meta-Sistema de EMC, está compuesto por el Meta-SMC y las diferentes plataformas multisistema. Cada SMC de las diferentes plataformas multisistema comparte su información con el Meta-SMC, de este modo, el Meta-SMC centraliza toda la información de mantenimiento de las diferentes plataformas del Meta-Sistema.

Al igual que los EMC permiten incrementar el tiempo de operación de una plataforma, minimizando las tareas de mantenimiento y por lo tanto reduciendo costes, al aplicar el concepto de Meta-Sistema de EMC estas ventajas se multiplican facilitando una gestión integral del mantenimiento.

Un ejemplo de un sector en el cual está comenzando a ser característico la utilización de Meta-Sistemas de EMC, son las aerolíneas.

Analizando el ejemplo de un Meta-EMC de una aerolínea, de acuerdo a la Figura 21, los aviones que dispongan de un SMC integrado, son cada uno de los EMC, y el Meta-SMC forma parte del centro de gestión y mantenimiento de la aerolínea. De este modo, durante la operación de la flota de aviones, el centro de mantenimiento de la aerolínea a través del Meta-SMC, recibe en tiempo real la información de mantenimiento de todos sus aviones. Con toda la información de mantenimiento centralizada en tiempo real, la aerolínea tiene la capacidad de gestionar de forma eficiente las tareas de mantenimiento, incrementando las siguientes ventajas:

- Planificar las paradas de mantenimiento,
- Minimizar el tiempo que los aviones están en mantenimiento.
- Gestionar de forma eficaz de la mano de obra especializada.
- Gestión eficaz de repuestos.

Todas estas ventajas contribuyen a maximizar las capacidades de la flota, permitiendo una gestión eficaz de aerolínea, lo cual se traduce en un incremento de los beneficios.

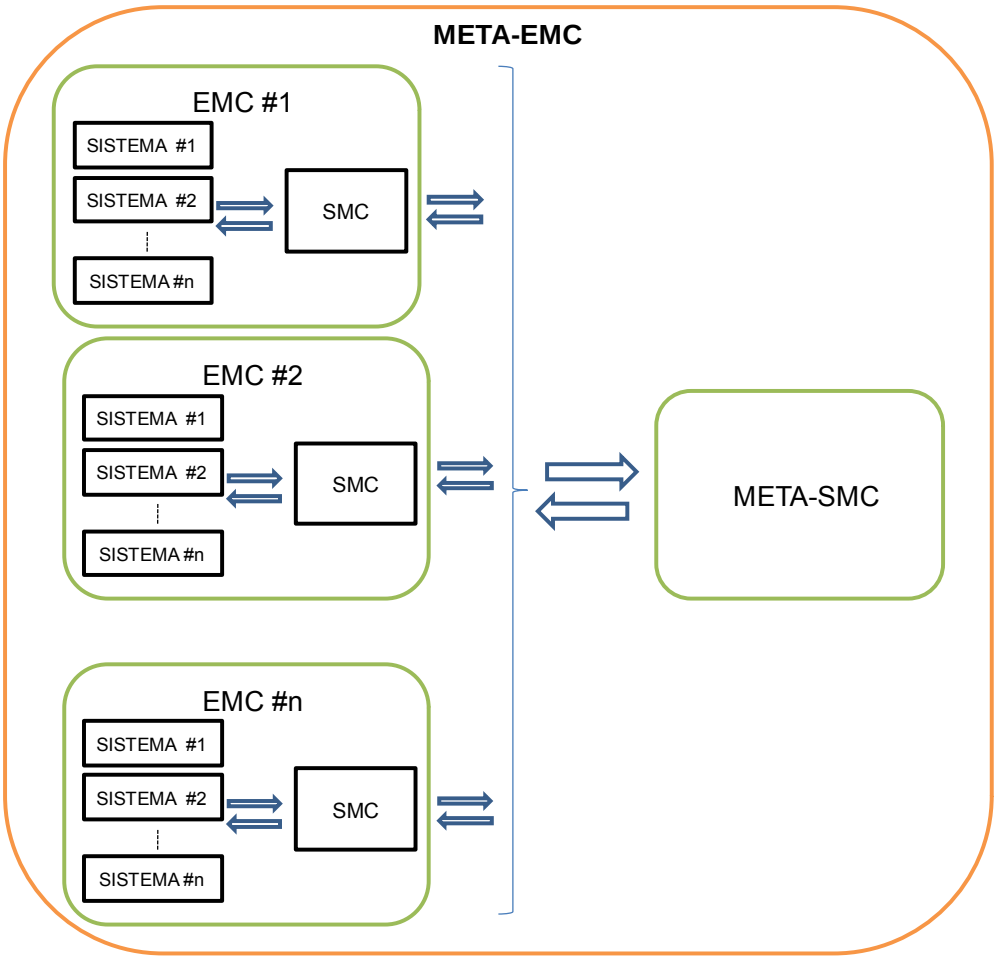


Figura 21. Esquema de una Arquitectura Meta-sistema para EMC.

5.6. Predicción de fallos basados en la distribución de probabilidad Weibull

Partiendo de toda la información almacenada en el SMC de los fallos de los diferentes sistemas de una plataforma multisistema, este TFM propone el procesado de dicha información con el fin de predecir el tiempo de vida de los equipos de la plataforma.

Para el cálculo de las predicciones de fallos o del tiempo de vida de los equipos se ha seleccionado la distribución de probabilidad de Weibull, debido a la flexibilidad y versatilidad que ofrece al adaptarse a las distintas tendencias que puede presentar la distribución de fallos de un sistema, sin necesidad de realizar un pre análisis de los datos [32].

5.6.1. Distribución de probabilidad Weibull

Aunque la distribución de probabilidad de Weibull fue desarrollada anteriormente por otros matemáticos, debe su nombre al matemático sueco que documentó esta distribución de probabilidad en 1939.

La principal bondad de esta distribución de probabilidad se encuentra en su gran adaptación a otras diferentes distribuciones de probabilidad como exponencial, gaussiana,… en función de sus parámetros característicos [32].

De este modo, en base al registro de fallos de un equipo, la distribución de probabilidad de Weibull posibilita conocer la distribución de fallos de dicho equipo y predecir su tiempo de vida.

Los cálculos de probabilidad de la distribución de Weibull, se basan en la caracterización de los siguientes parámetros:

- t_0: parámetro de localización, indica el origen del tiempo.
- β: parámetro de forma, define la monotonía de la razón de fallos, creciente, constante o decreciente.
- η: parámetro de escala, define la escala de la distribución en el eje t.

Aunque la obtención algebraica de los parámetros característicos β y η de la distribución de Weibull, requieren de un procesado matemático complejo, como el método de los momentos o el método de máxima verosimilitud, existe un método gráfico que simplifica los cálculos, ver apartado 5.6.2.

En los siguientes apartados se presentan las distintas ecuaciones de la distribución de probabilidad de Weibull en función de sus parámetros característicos.

5.6.1.1. Función de densidad de probabilidad, f(t)

La función de densidad de probabilidad, f(t), (1), calcula la probabilidad relativa de que un evento se produzca en un cierto instante de tiempo. Para la aplicación de la distribución de Weibull al EMC, los eventos son los fallos de los equipos.

La Figura 22 muestra las gráficas de la función de densidad de probabilidad en función de distintos valores de β.

$$f(t) = \begin{cases} \frac{\beta}{\eta}\left(\frac{t-t_0}{\eta}\right)^{\beta-1} e^{-\left(\frac{t-t_0}{\eta}\right)^{\beta}}, & t \geq 0 \\ 0, & t < 0 \end{cases} \qquad (1)$$

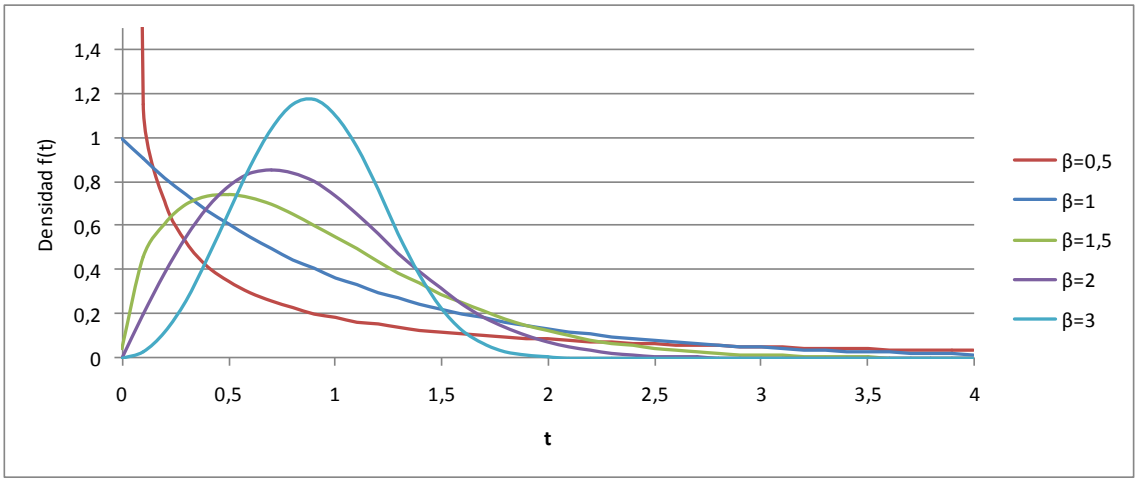


Figura 22. Función densidad de probabilidad por Weibull.

5.6.1.2. Función distribución de probabilidad, F(t)

La función de distribución de probabilidad, F(t), (2), calcula la probabilidad acumulada de que un evento se produzca en un cierto tiempo. La Figura 23, muestra las gráficas de la función de distribución de probabilidad en función de distintos valores de β.

$$F(t) = \begin{cases} 1 - e^{\left(\frac{t-t_0}{\eta}\right)^{\beta}}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{2}$$

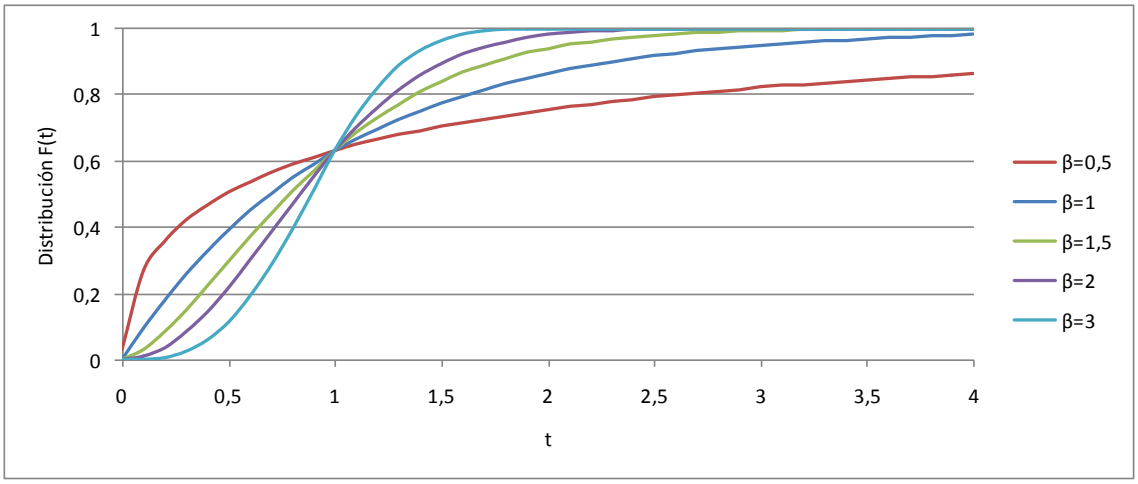


Figura 23. Distribución de probabilidad por Weibull.

5.6.1.3. Fiabilidad, R(t)

La función de fiabilidad, R(t), (3), (4) y (5), calcula la probabilidad de que un equipo sobreviva a un instante de tiempo. La Figura 24, muestra la variación de las diferentes gráficas de la función de fiabilidad en función de distintos valores de β.

$$R(t) = 1 - F(t) \tag{3}$$

$$R(t) = 1 - \left(1 - \left(\frac{t-t_0}{\eta}\right)^{\beta}\right) \tag{4}$$

$$R(t) = e^{-\left(\frac{t-t_0}{\eta}\right)^{\beta}} \tag{5}$$

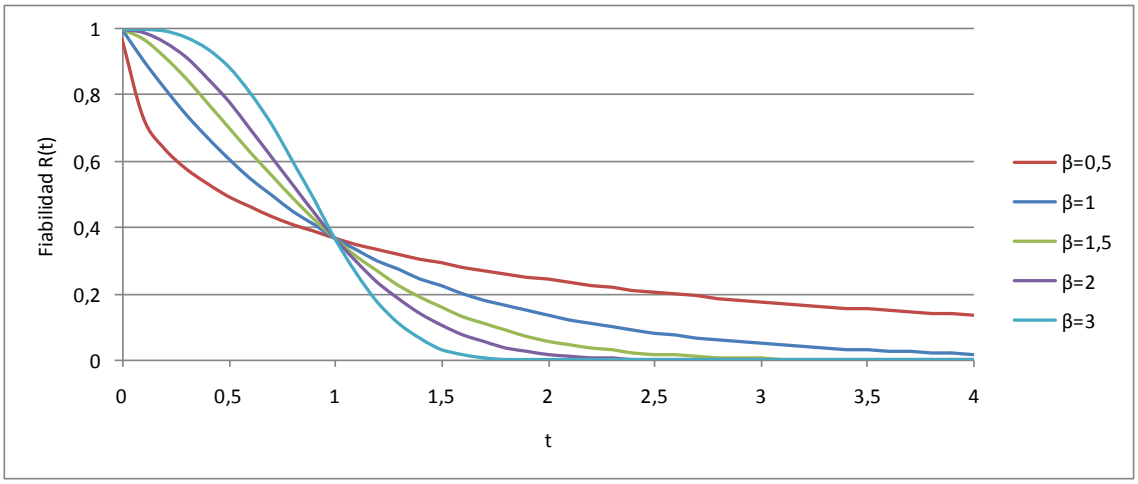


Figura 24. Función Fiabilidad por Weibull.

5.6.1.4. Tasa de fallos, λ(t)

La función de la tasa de fallos, λ(t) (6) y (7), representa como se distribuyen los fallos que presenta un equipo en función del tiempo. La Figura 25 muestra las diferentes tendencias de la tasa de fallo en función del valor de β, mientras que la Figura 26 presenta las diferentes zonas de la curva de la bañera en función del tiempo (t).

$$\lambda(t) = \frac{f(t)}{R(t)} \tag{6}$$

$$\lambda(t) = \begin{cases} \frac{\beta}{\eta}\left(\frac{t-t_0}{\eta}\right)^{\beta-1}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{7}$$

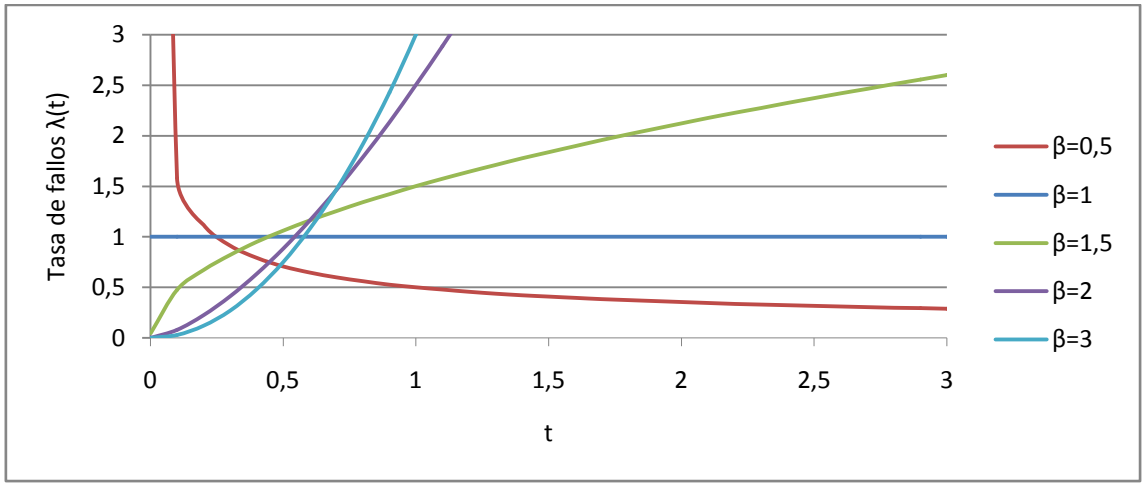


Figura 25. Tasa de fallos de Weibull.

En función del valor del parámetro de forma, β, la tasa de fallos de Weibull se aproxima a las diferentes zonas características de una curva de la bañera, Figura 26.

- Si β < 1, determina la zona de mortalidad infantil de un equipo, en la cual la tasa de fallos disminuye con la edad del equipo, sin llegar a cero.

- Si β = 1, determina la zona de vida útil del equipo, en la cual la tasa de fallo se mantiene constante, con un valor distinto de cero, debido a los fallos aleatorios del equipo. Para β = 1, la distribución de probabilidad de Weibull es similar a una distribución de probabilidad exponencial.

- Si β > 1, determina la zona de envejecimiento o desgaste del equipo, en la cual, la tasa de fallo se incrementa con la edad del equipo.

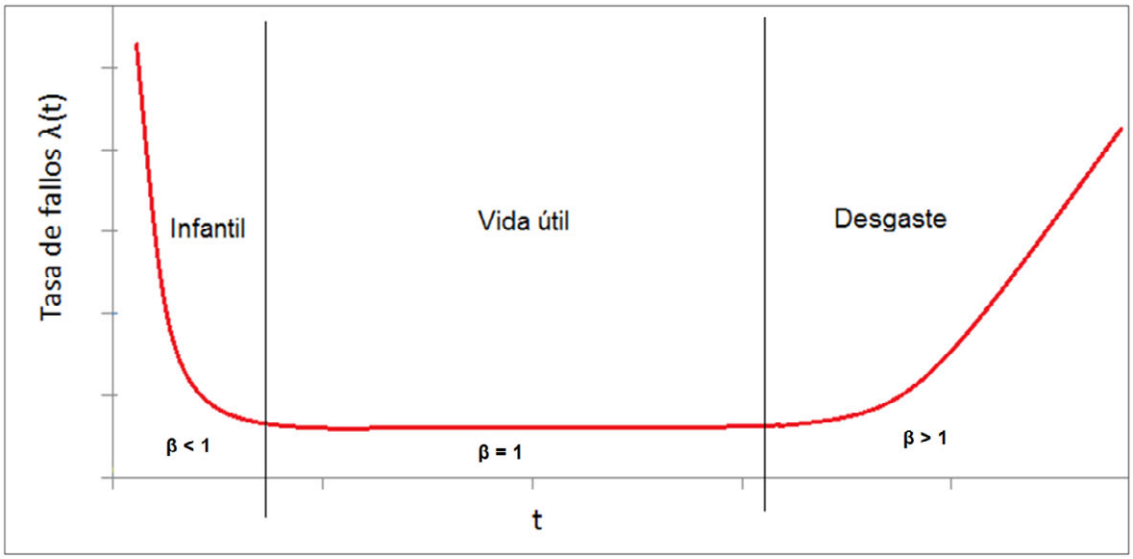


Figura 26. Curva de la bañera.

5.6.1.5. Tiempo medio entre fallos (MTBF)

El tiempo medio entre fallos (MTBF), define la esperanza de vida media de un equipo, (8). El MTBF se define para equipos que pueden ser reparados,

en caso contrario, si el equipo se descarta tras un fallo, este parámetro se denomina tiempo medio hasta el fallo (MTTF).

$$MTBF = \eta\Gamma(1 + \frac{1}{\beta}) \quad (8)$$

5.6.2. Aplicación de la distribución Weibull en la predicción de fallos.

A partir del registro de fallos de un equipo, sabiendo que el equipo ha sido sustituido cada vez que ha fallado, por medio de la distribución de Weibull se puede determinar el tiempo de vida medio del equipo y cuándo debería ser sustituido en función de la fiabilidad deseada.

El siguiente ejemplo, desarrolla el cálculo de la distribución de Weibull a partir de un registro de fallos de un equipo, Tabla III, ordenado por horas de fallo, siendo N el número total de fallos.

Fallo Nº	Horas
1	1218
2	1354
3	1521
4	1560
5	1624
6	1720
7	1780
8	1815
9	2413
10	2710

Tabla III. Registro de fallos del sistema bajo estudio.

Calculando la probabilidad de cada fallo F(t), *(9)*, en función del número total de fallos más uno. Se añade uno al número total de fallos dado que no se descarta que otro equipo dure más horas que el equipo que más horas ha durado del registro actual.

$$\mathrm{F(t)} = \mathrm{Fallo\ Nº}/(\mathrm{N}+1) \quad (9)$$

Fallo Nº	Horas	F(t)
1	1218	0,091
2	1354	0,182
3	1521	0,273
4	1560	0,364
5	1624	0,455
6	1720	0,545
7	1780	0,636
8	1815	0,727
9	2413	0,818
10	2710	0,909

Tabla IV. Distribución de probabilidad de fallo F(t).

El siguiente paso es calcular los parámetros característicos de la distribución de Weibull β y η.

5.6.2.1. Método gráfico para el cálculo de los parámetros β y η

El método gráfico para el cálculo de los parámetros β y η se basa en la representación gráfica de los fallos de un equipo utilizando el gráfico de Weibull. Las características de este gráfico de Weibull están en los ejes. En el eje de ordenadas representa la función (10), mientras que el eje de abscisas representa la función (11).

$$\text{Eje de ordenadas: } ln\left(ln\left[\frac{1}{1-F(t)}\right]\right) \quad (10)$$

$$\text{Eje de abscisas: } ln(t - t_0) \quad (11)$$

Calculando el valor de los ejes en función de las ecuaciones (10) y (11)

Una vez representados los fallos del equipo en el gráfico de Weibull trazamos una recta de tendencia por los puntos indicados en el gráfico.

Fallo Nº	Horas	F(t)	LnLn(1/(1-F(t)))	Ln(t)
1	1218	0,091	-2,351	7,105
2	1354	0,182	-1,606	7,211
3	1521	0,273	-1,144	7,327
4	1560	0,364	-0,794	7,352
5	1624	0,455	-0,500	7,392
6	1720	0,545	-0,237	7,450
7	1780	0,636	0,011	7,484
8	1815	0,727	0,261	7,503
9	2413	0,818	0,533	7,788
10	2710	0,909	0,874	7,904

Tabla V. Cálculo de puntos para el gráfico de Weibull.

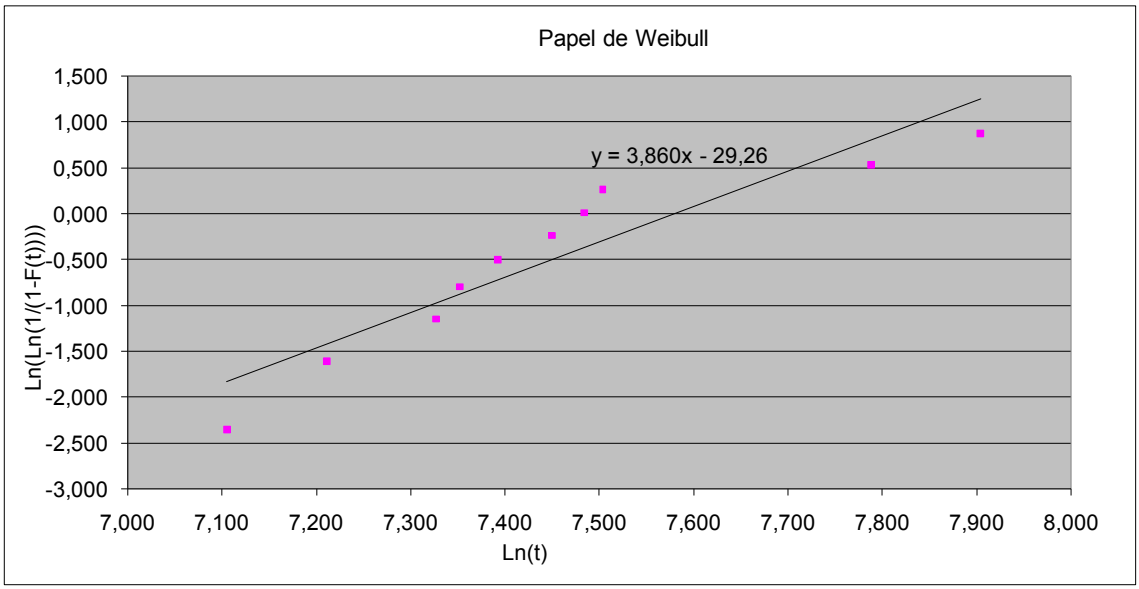


Figura 27. Gráfico de Weibull.

A partir de los valores de la recta de tendencia de los puntos presentados en el gráfico de Weibull, se calcula β y η, de acuerdo a la ecuación (12).

$$y = ax + b = \beta x - \beta \ln \eta \rightarrow \begin{cases} a = \beta \\ b = -\beta \ln \eta \end{cases} \quad (12)$$

Sustituyendo los coeficientes de la recta de tendencia en la ecuación (12), obtenemos el valor de los parámetros característicos: β= 3,860 y η= 1959,24.

5.6.2.2. Densidad, Probabilidad, Fiabilidad, Tasa de fallos y MTBF

Con los valores calculados para β y η, aplicando en las ecuaciones (1), (2), (5), y (7) obtenemos las gráficas de distribución de densidad, Figura 28, distribución de probabilidad, Figura 29, distribución de fiabilidad, Figura 30 y Tasa de fallos, Figura 31.

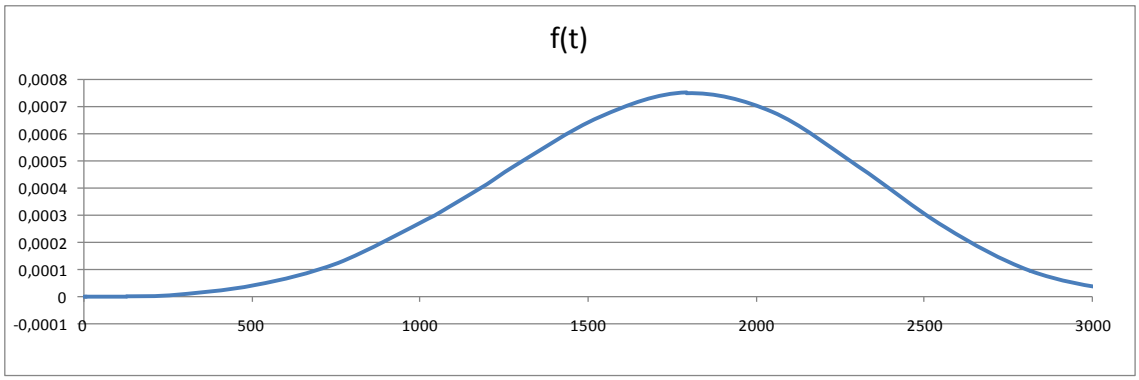


Figura 28. Distribución de densidad, f(t).

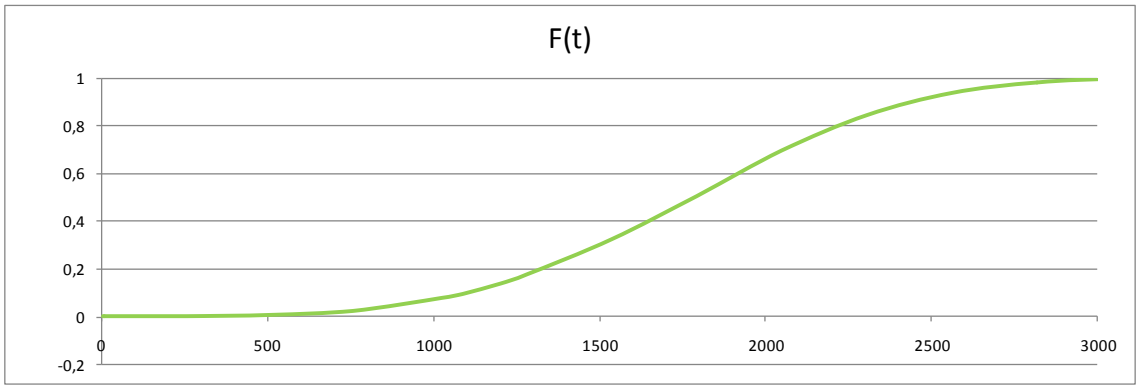


Figura 29. Distribución de probabilidad, F(t).

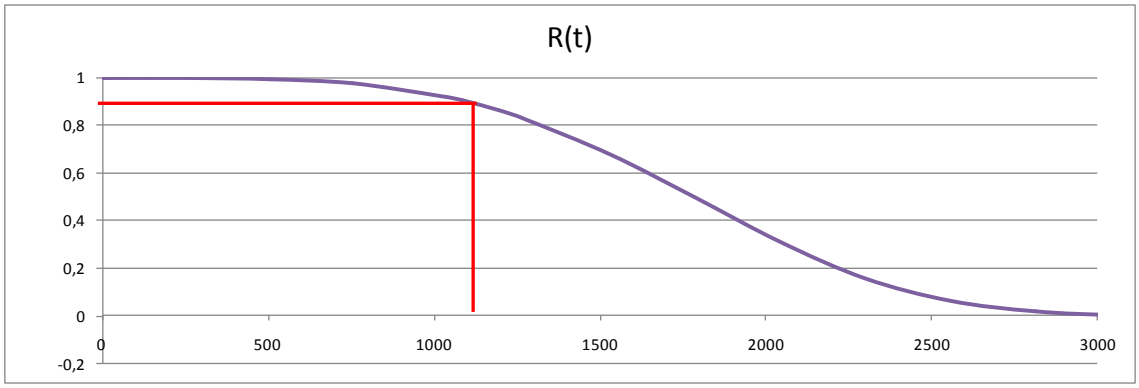


Figura 30. Distribución de fiabilidad, R(t).

Según la gráfica de la función de la distribución de fiabilidad R(t), el equipo analizado tendría una fiabilidad del 90% para 1085 horas de funcionamiento.

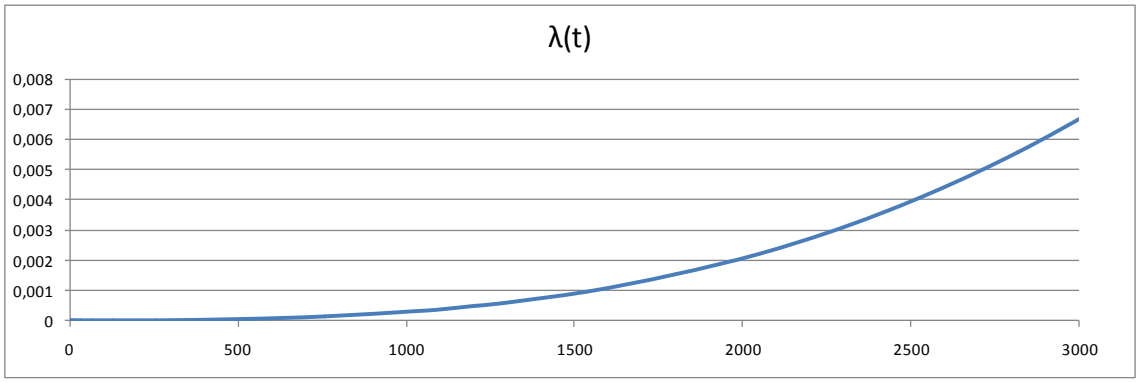


Figura 31. Tasa de fallos, λ(t).

Dado que β > 1, indica que el equipo está en zona de desgaste o envejecimiento.

Por último el MTBF del equipo, aplicando a la ecuación (8) los valores obtenidos de β y η, tiene un valor de 1772,28 horas.

5.7. Mensajes de fallos espurios

En ocasiones, las funciones de detección de fallos de los sistemas pueden generar mensajes de fallo que no corresponden a ningún fallo real del sistema. Estos mensajes, son reportados al SMC y por lo tanto requieren por parte del operador de una acción de mantenimiento, la cual será innecesaria, dado que no existe ningún fallo real en el sistema.

Estos mensajes de fallo que no requieren acciones de mantenimiento, son denominados fallos espurios.

Un sistema puede propagar un mensaje de fallo, que dependiendo de ciertas condiciones del sistema, puede ser un mensaje de fallo espurio, y en otras condiciones puede ser real.

La corrección de los mensajes de fallo espurios en el diseño de un sistema, en muchas ocasiones, es una tarea complicada, costosa y que no siempre es viable. Por lo tanto, cuando el diseño de un sistema está finalizado, los diseñadores de dicho sistema, deben informar de si el sistema diseñado, a través de su interfaz de mantenimiento, propaga algún fallo espurio, y si estos fallos son espurios en ciertas condiciones o si por el contrario, son fallos espurios siempre que son propagados.

Esta información sobre los mensajes de fallos espurios de todos los sistemas de una plataforma, debe ser recogida y publicada a los operadores de mantenimiento de una plataforma, con el fin de evitar la realización de tareas de mantenimiento innecesarias.

Adicionalmente, hay mensajes de fallo, que tras finalizar el diseño de un sistema no han sido declarados como mensajes de fallo espurios y que con la experiencia en operación del sistema, se ha demostrado que son fallos espurios siempre que son propagados, o en determinadas condiciones. Por lo tanto, la gestión de los mensajes espurios, es una tarea que puede permanecer en el tiempo, posteriormente a la finalización del diseño de los sistemas.

5.7.1. Filtro de fallos espurios

Teniendo en cuenta que los principales objetivos de un EMC son conseguir un mantenimiento óptimo y eficaz, los mensajes de fallos declarados como espurios para todas las circunstancias en las que sean propagados, no deberían ser reportados al SMC.

Debido a que la corrección de los mensajes de fallo espurios por parte de los sistemas no es factible en la mayoría de los diseños, una solución para evitar que estos mensajes, sean mostrados por el SMC al operador de mantenimiento, es la integración en el SMC de un filtro de mensajes de fallos espurios.

De este modo, el SMC no presentará al operador de mantenimiento, los mensajes de fallo espurios, evitando la realización de tareas de mantenimiento innecesarias, o la comprobación de si el mensaje de fallo en cuestión está declarado como espurio.

5.8. Beneficios e inconvenientes de un EMC

Si bien a lo largo del TFM se han desarrollado algunas de los beneficios que supone la integración de un EMC en una plataforma multisistema, en este punto, desglosa los beneficios e inconvenientes de los EMC, [29].

5.8.1. Beneficios de un EMC

La integración de los sistemas de una plataforma en un EMC, suponen una serie de beneficios a diferentes niveles.

- A nivel sistema, la incorporación de funciones BITE en los sistemas, evitan en gran parte la necesidad de realizar tareas de mantenimiento que requieran la utilización de equipos de prueba externos, para estimular o monitorizar al sistema, y por lo tanto se reducen las interfaces del sistema con el exterior.

Esto lleva consigo, que se realicen menos desmontajes innecesarios del sistema, evitando así los posibles problemas derivados de la manipulación del sistema. Además la detección de una anomalía es detectada de forma rápida, reduciendo el tiempo de intervención.

Adicionalmente, la monitorización de los mensajes de fallo pueden ser utilizados para una futura mejora del sistema.

- A nivel plataforma, la monitorización continua de todos los sistemas, permite que la detección de fallos sea rápida y eficiente. Esto permite la reducción del tiempo necesario para desarrollar las tareas de mantenimiento, lleva consigo que la plataforma deba estar menos tiempo inoperativa.

  Además, como el SMC realiza los diagnósticos de forma precisa, las tareas de mantenimiento a realizar son conocidas, y por lo tanto no es necesario personal de mantenimiento altamente capacitado.

Todas estas ventajas, se resumen en una reducción del coste del ciclo de vida de la plataforma.

5.8.2. Inconvenientes

La implementación de un EMC en una plataforma, aparte de introducir múltiples beneficios, también lleva consigo algunos inconvenientes, que deben ser conocidos de antemano con el fin de minimizar sus efectos.

- A nivel sistema, la implementación de las capacidades BITE compatibles con un EMC, hace que el diseño y desarrollo de los sistemas sea más complejo, y requiera la incorporación de hardware y software adicional. Con lo cual aparte de incrementar el coste, se dilata el tiempo de diseño y desarrollo del sistema.

- A nivel plataforma, la implementación de un EMC, aparte del propio coste del EMC, supone incorporar en la plataforma, nuevos equipos, cableados, etc., lo cual implica un incremento de

consumo eléctrico, peso, espacio, que para algunas plataformas, con por ejemplo una aeronave, son factores que deben ser tenidos en cuenta.

- Desde el punto de vista de operación de mantenimiento, en el EMC, como explica el apartado 5.7, los sistemas pueden propagar códigos de fallo espurios, incluso puede darse la circunstancia que tras la aparición de un código de fallo, el fallo indicado no sea reproducible (NFF).

6 Implementación de un Entorno de Mantenimiento Centralizado modelado en LabVIEW

6.1. Entorno de simulación

Con el objetivo de seleccionar la herramienta adecuada sobre la cual desarrollar el EMC primero se han establecido los principales requisitos necesarios para implementar el modelo de EMC:

- Simulación simultanea de diferentes sistemas en una única máquina.
- Protocolo de comunicaciones multi-punto.
- Gestión y almacenamiento de datos.
- Generación de informes.
- Desarrollo de interfaces gráficas de usuario.

Tras realizar el análisis de estos requisitos y teniendo en cuenta otros posibles requisitos adicionales como la posibilidad de ampliación del modelo EMC, incluso la capacidad de utilizar el modelo de EMC con algún prototipo real de los equipos que componen el modelo, la herramienta seleccionada ha sido LabVIEW.

El desarrollo y simulación del modelo de EMC se ha realizado con LabVIEW versión 13.0.1 (32-bit).

6.2. Descripción del Modelo en LabVIEW del Entorno Mantenimiento Centralizado

El modelo del EMC ha sido definido de forma genérica. Por lo tanto, la definición del modelo de la plataforma multisistema está basada en un conjunto variable de sistemas genéricos, junto a un modelo de un SMC. La Figura 32 presenta el esquema de la arquitectura utilizada en el modelo de EMC.

Para modelar los sistemas genéricos del EMC se utilizado un único modelo de sistema, que ha sido denominado Sistema Patrón. Este Sistema Patrón se duplica tantas veces como sistemas genéricos se quieran tener en la

simulación, con un límite de 10 sistemas. Cada Sistema Patrón se identifica de forma única con el fin de que el SMC sea capaz de identificarlo de forma univoca.

El modelo del Sistema Patrón, permite al usuario simular fallos, los cuales serán procesados y enviados a través de un protocolo de comunicaciones al modelo de SMC.

El modelo del SMC integra las principales funciones descritas en los apartados anteriores y mostradas en la Figura 13. Para ello se ha definido una interfaz de usuario, a través de la cual se accede a los diferentes modos de funcionamiento y funciones del SMC.

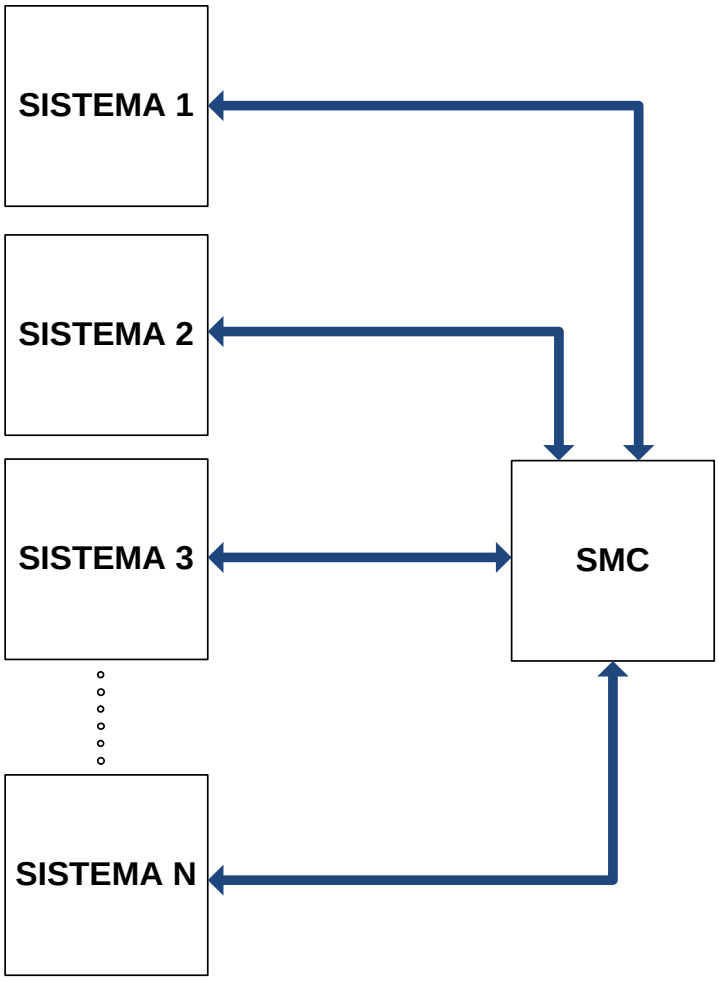


Figura 32. Esquema del modelo en LabVIEW del EMC.

6.3. Arquitectura del modelo

La arquitectura del modelo de EMC, se puede esquematizar de forma análoga a un esquema de capas OSI [21], [22], como muestra la Figura 33.

Esta figura muestra las diferentes capas que componen los tres bloques principales el EMC: SMC, Red y Sistemas Patrón. Desde la capa inferior, que representa el nivel físico, formada por los tres módulos de cada Sistema Patrón, hasta llegar a la capa superior la cual representa las diferentes funciones del SMC.

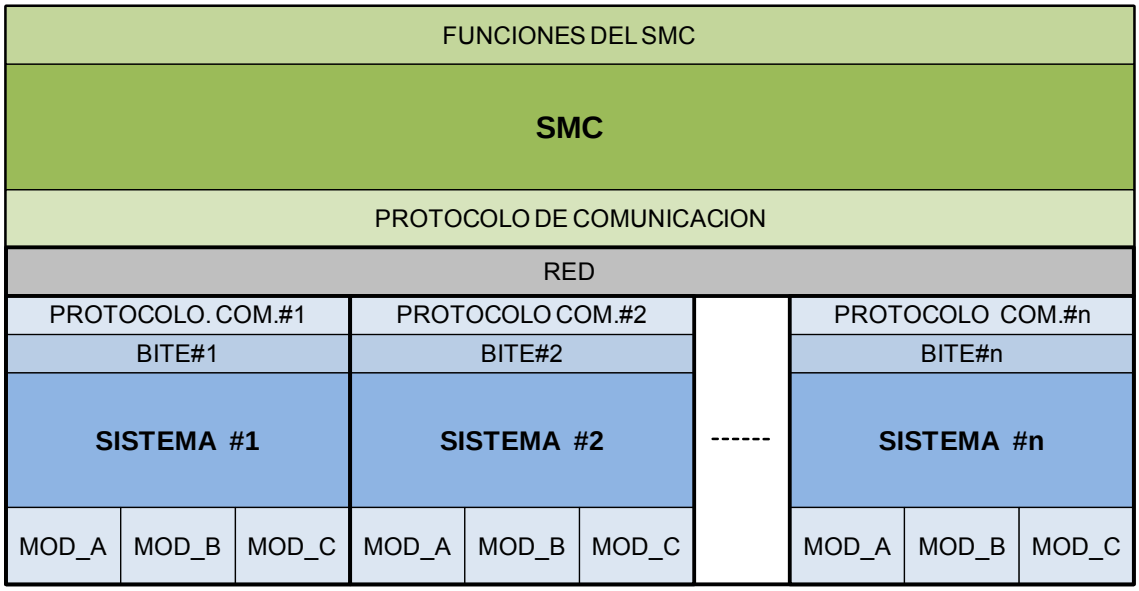


Figura 33. Esquema de la arquitectura del modelo EMC.

6.3.1. Sistema Patrón

La arquitectura del modelo del Sistema Patrón se ha definido para cumplir con los siguientes objetivos:

- Simular fallos en los distintos módulos del Sistema Patrón.
- Monitorizar periódicamente las posibles condiciones de fallo del sistema.
- Enviar la información de los fallos registrados en el sistema.

- Recibir comandos del SMC.

Para cumplir con estos objetivos, la arquitectura implementada en el modelo de LabVIEW implementa los siguientes bloques:

- Interfaz de usuario
- Monitorización de fallos y procesamiento de datos.
- Recepción y envío de datos al SMC vía TCP/IP [33].
- Gestión de errores.

La interfaz de usuario permite simular y monitorizar fallos en los cuatro módulos modelados para el Sistema Patrón, ver la Figura 34.

- Fuente de alimentación.
- Módulo Entrada / Salida.
- Procesador.
- Sensor / Actuador y Unidad de Control / Indicador.

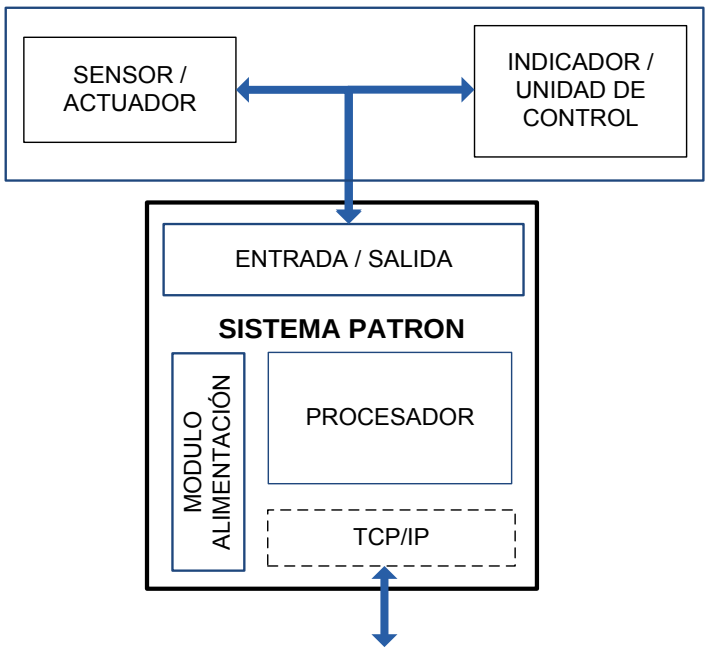


Figura 34. Esquema de la arquitectura de un Sistema Patrón.

6.3.2. Sistema de Mantenimiento Centralizado

El modelo del SMC se ha diseñado con la siguiente arquitectura, ver Figura 35, con el fin de cumplir con los siguientes objetivos:

- Comunicación vía TCP/IP con todos los Sistemas Patrón del EMC.
- Análisis de los datos recibidos de todos los Sistemas Patrón.
- Almacenamiento de los fallos recibidos en archivos XML.
- Presentación del estado de los sistemas.
- Procesado de datos almacenados en XML para las funciones mantenimiento.

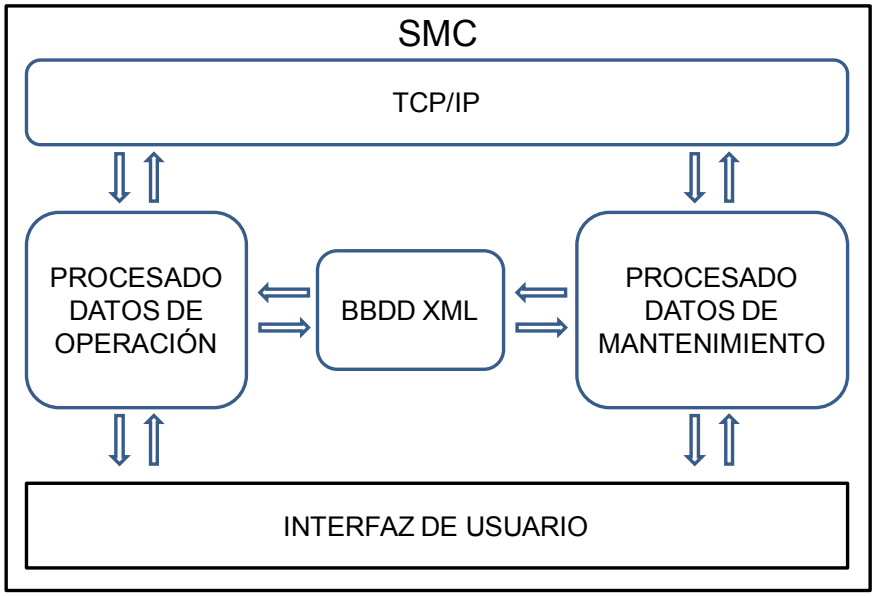


Figura 35. Esquema de la arquitectura del SMC.

6.3.3. Protocolo de comunicación

Los principales requisitos para definir la arquitectura del sistema de comunicación entre los diferentes Sistemas Patrón y el SMC son:

- El SMC debe tener comunicación con los diferentes Sistemas Patrón.
- Comunicación bidireccional.

En base a estos requisitos y teniendo en cuenta los diferentes tipos de comunicación que permite modelar LabVIEW, se ha optado por la utilización del protocolo TCP/IP.

En la definición del protocolo TCP/IP en LabVIEW, el SMC actúa como servidor y los diferentes Sistemas Patrón actúan como clientes [34], [35], [36].

La Figura 36 muestra la secuencia TCP/IP establecida en el modelo LabVIEW para el Sistema Patrón (cliente), esta secuencia se ejecuta después de cada monitorización.

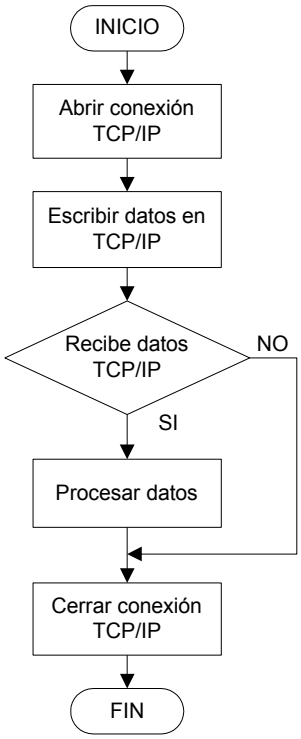


Figura 36. Secuencia TCP/IP en Sistema Patrón.

La Figura 37 muestra la secuencia TCP/IP establecida en LabVIEW para el SMC (servidor).

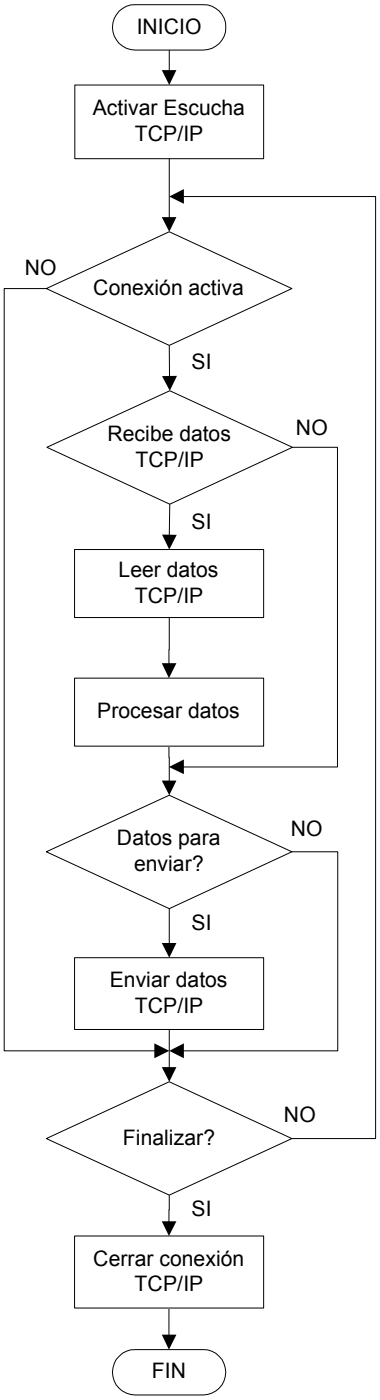


Figura 37. Secuencia TCP/IP en SMC.

6.3.4. Almacenamiento de datos

Un objetivo fundamental de un EMC es el almacenamiento de la información recibida de cada sistema. Esta información se debe almacenar de forma ordenada, con el fin de que se pueda acceder de forma rápida y eficaz a la información necesaria en cada momento.

En la implementación del modelo del EMC en LabVIEW, para el almacenamiento de información por parte del SMC, se ha optado por archivos XML [37].

El SMC gestiona un archivo XML por cada sistema de la plataforma en el cual almacena la información de acuerdo a un patrón establecido.

La Tabla VI muestra los campos que almacenada el SMC para cada fallo.

Fecha y hora del fallo
Sistema que reporta el fallo
Número de parte del sistema
Número de serie del sistema
Número de test
Código de fallo
Módulo acusado 1
Módulo acusado 2
Módulo acusado 3

Tabla VI. Paquete de información por fallo almacenado en XML.

La estructura de los archivos XML para cada sistema se muestra en la Figura 38.

```
▼<Event>
  ▼<DATOS_FALLOS>
    ▼<FALLO>
       <FECHA-HORA>17/04/2015 15:53:41</FECHA-HORA>
       <SISTEMA>SIST_1</SISTEMA>
       <NUMERO_PARTE>S1_001</NUMERO_PARTE>
       <NUMERO_SERIE>A00001</NUMERO_SERIE>
       <NUMERO_TEST>CBIT</NUMERO_TEST>
       <CODIGO_FALLO>1</CODIGO_FALLO>
       <MOD_ACUSADO_1>Mod_A</MOD_ACUSADO_1>
       <MOD_ACUSADO_2>Mod_B</MOD_ACUSADO_2>
       <MOD_ACUSADO_3>N/A</MOD_ACUSADO_3>
     </FALLO>
     ┆
     ┆
     ┆
     ┆
     ┆
    ▼<FALLO>
       <FECHA-HORA>06/05/2015 21:26:59</FECHA-HORA>
       <SISTEMA>SIST_1</SISTEMA>
       <NUMERO_PARTE>NP_SP1_001</NUMERO_PARTE>
       <NUMERO_SERIE>NS_SP1_001</NUMERO_SERIE>
       <NUMERO_TEST>CBIT</NUMERO_TEST>
       <CODIGO_FALLO>1</CODIGO_FALLO>
       <MOD_ACUSADO_1>Mod_A</MOD_ACUSADO_1>
       <MOD_ACUSADO_2>Mod_B</MOD_ACUSADO_2>
       <MOD_ACUSADO_3>N/A</MOD_ACUSADO_3>
     </FALLO>
  </DATOS_FALLOS>
</Event>
```

Figura 38. Formato del contenido de los archivos XML.

6.4. Funciones del modelo de EMC

En la implementación del modelo del EMC se han desarrollado e implementado diferentes funciones, para el Sistema Patrón y para el SMC.

6.4.1. Funciones del Sistema Patrón.

El modelo del Sistema Patrón implementa los tres modos de funcionamiento del BITE de un sistema, Test de Inicio, Test Continuo y el Test Interactivo, por medio de las funciones desarrolladas a continuación.

6.4.1.1. Simulación de fallos en el Sistema Patrón

La función principal del modelo del Sistema Patrón es simular diferentes fallos en los distintos módulos del sistema, ver Figura 34. Para ello se ha diseñado la interfaz de usuario, mostrada en la Figura 39.

Figura 39. Interfaz de usuario del Sistema Patrón.

La interfaz de usuario del Sistema Patrón está dividida en 3 zonas, como indica la Figura 40, en la zona superior marcada en azul, están situados los campos de identificación del sistema, la zona izquierda, marcada en verde, se sitúan los controles para la simulación de fallos en los cuatro módulos en los que se divide el Sistema Patrón. En la zona derecha, marcada en naranja, aparecen las indicaciones de los fallos simulados para cada módulo.

Figura 40. Campos de la interfaz de usuario del Sistema Patrón.

El Sistema Patrón debe permitir simular manualmente fallos para los cuatro módulos en los que está dividido.

- Alimentación:

  El Sistema Patrón se ha modelado con dos alimentaciones de tensión continua de 28VDC+/-2VDC. Los controles para la simulación de fallos en el módulo de la alimentación permiten variar los valores de tensión, el nivel de rizado y provocar interrupciones, en ambas alimentaciones.

- Entrada / Salida:

  Este módulo se ha modelado con entradas discretas y analógicas, y salidas discretas y analógicas, junto con 4 buses. Los controles permiten simular un fallo en cualquiera de las entradas salidas y buses.

- Procesador:

  En el módulo de control de fallo del procesador, se han modelado un control de la temperatura y otro del rendimiento del procesador, junto a seis controles para simular los siguientes fallos: escritura RAM, lectura RAM, lectura ROM, fallo microprocesador, fallo 001, fallo 002.

- Sensor / Actuador y Unidad de Control:

  Para el módulo Sensor / Actuador y Unidad de Control, se han modelado un control de rango para sensor y actuador junto con seis controles para simular los siguientes fallos: fallo sensor, fallo actuador, fallos de unidad de control: 1, 2, 3 y 4.

En la Tabla VII, se presenta los fallos generados a través de estos controles.

Cuando en el Sistema Patrón se activa un fallo, se presenta mediante un Led indicador en la parte derecha de la interfaz.

6.4.1.2. Lista de fallos del sistema

La definición de cada fallo, indica las condiciones de fallo respecto a las condiciones nominales de la señal, módulo o función que se esté tratando.

Módulo	Descripción del fallo	Código de fallo	Condición de fallo
Fuente de alimentación	VDC#1 excesivo	1	VDC#1 > 30VDC
	VDC#1 insuficiente	2	VDC#1 < 26VDC
	Rizado VDC#1 excesivo	3	Rizado#1 > 5%
	Interrupción VDC#1	4	Interrupción VDC#1 activo
	VDC#2 excesivo	5	VDC#2 > 30VDC
	VDC#2 insuficiente	6	VDC#2 < 26VDC
	Rizado#2 excesivo	7	Rizado#2 > 5%
	Interrupción VDC#2	8	Interrupción VDC#2 activo
Entrada / Salida	Entradas discretas	9	Alguna de las 8 entradas falladas
	Salidas discretas	10	Alguna de las 8 salidas falladas
	Entradas analógicas	11	Alguna de las 8 entradas falladas
	Salidas analógicas	12	Alguna de las 8 salidas falladas
	Bus 1	13	Fallo Bus 1 activo
	Bus 2	14	Fallo Bus 2 activo
	Bus 3	15	Fallo Bus 3 activo
	Bus 4	16	Fallo Bus 4 activo
Procesador	Temp. Procesador	17	Temp. Procesador > 80ºC
	Rendimiento Procesador	18	Rendimiento procesador > 90 %
	Fallo Procesador	19	Fallo Procesador activo
	Fallo Escritura RAM	20	Fallo Escritura RAM activo
	Fallo Lectura RAM	21	Fallo Lectura RAM activo
	Fallo ROM	22	Fallo ROM activo
	Fallo 001	23	Fallo 001 activo
	Fallo 002	24	Fallo 002 activo
Sensor / Actuador	Sensor fuera de rango	25	Sensor >100% o <0%
	Actuador fuera de rango	26	Actuador >100% o <0%
	Fallo actuador	27	Fallo actuador activo
	Fallo sensor	28	Fallo sensor activo
Indicador / Unidad de control	Fallo 1 unidad de control	29	Fallo 1 unidad de control activo
	Fallo 2 unidad de control	30	Fallo 2 unidad de control activo
	Fallo 3 unidad de control	31	Fallo 3 unidad de control activo
	Fallo 4 unidad de control	32	Fallo 4 unidad de control activo

Tabla VII. Tabla de fallos del Sistema Patrón.

6.4.1.3. Monitorización y envío de fallos al SMC

El Sistema Patrón, de forma periódica monitoriza el estado del sistema a una frecuencia de 1Hz.

Durante la monitorización se comprueban si alguno de los posibles fallos está activo. Con el resultado de la monitorización se completa el paquete de información que muestra la Tabla IX. Para los fallos que no estén activos, el contenido de los campos Código de fallo es “0” y para los Módulos Acusados es “N/A”.

El primer ciclo que realiza el Sistema Patrón tras su puesta en marcha, corresponde al Test de Inicio por lo que el campo Modo de test se completa con “IBIT” (BIT Inicio). Para el resto de ciclos, este campo es “CBIT” (BIT Continuo).

Cada código de fallo, acusa a tres posibles módulos o equipos del sistema. El orden de los módulos acusados se establece según la probabilidad de que cada módulo sea el causante del fallo detectado. La Tabla VIII muestra los códigos de fallo con sus módulos acusados. Como ejemplo, el código de fallo 3 acusa como primera opción y probabilidad más alta al Módulo C y como segunda opción al Módulo A y no presentando ningún acusado en tercera probabilidad.

0	N/A	N/A	N/A
1	Mod_A	Mod_B	N/A
2	Mod_B	Mod_C	N/A
3	Mod_C	Mod_A	N/A
4	Mod_A	Mod_B	N/A
5	Mod_B	Mod_C	N/A
6	Mod_C	Mod_C	N/A
7	Mod_A	Mod_B	N/A
8	Mod_B	Mod_A	N/A
9	Mod_B	Mod_C	N/A
10	Mod_A	Mod_B	N/A
11	Mod_A	Mod_B	N/A
12	Mod_B	Mod_A	N/A
13	Mod_B	Mod_A	N/A
14	Mod_A	Mod_B	N/A
15	Mod_C	Mod_A	N/A
16	Mod_A	Mod_B	N/A
17	Mod_A	Mod_C	N/A
18	Mod_B	Mod_C	N/A
19	Mod_A	Mod_B	Mod_C
20	Mod_C	Mod_B	N/A
21	Mod_A	Mod_C	N/A
22	Mod_C	Mod_B	N/A
23	Mod_A	Mod_B	N/A
24	Mod_C	Mod_B	N/A
25	Mod_A	Mod_B	N/A
26	Mod_C	Mod_A	N/A
27	Mod_A	Mod_C	N/A
28	Mod_A	Mod_C	N/A
29	Mod_B	Mod_C	Mod_A
30	Mod_A	Mod_C	N/A
31	Mod_B	Mod_A	Mod_C
32	Mod_A	Mod_B	Mod_C
Fault Code	Affected MOD1	Affected MOD2	Affected MOD3

Tabla VIII. Códigos de fallo y módulos acusados.

El formato para los datos que se envían a través del protocolo TCP/IP establecido en LabVIEW entre los Sistemas Patrón y el SMC, es de tipo cadena de caracteres (*string*).

De acuerdo a la Tabla IX, cada Sistema Patrón, en cada ciclo de monitorización, envía un paquete de 132 datos de tipo cadena de caracteres.

Identificador del sistema
Número de parte
Número de serie
Modo de test
Código de fallo 1
Módulo acusado 1
Módulo acusado 2
Módulo acusado 3
Código de fallo 2
. . . .
Código de fallo 32
Módulo acusado 1
Módulo acusado 2
Módulo acusado 3

Tabla IX. Paquete de datos enviado periódicamente.

6.4.2. Funciones del Sistema de Mantenimiento Centralizado

En el modelo de SMC ha sido desarrollado con la arquitectura indicada en la Figura 35, permitiendo el acceso a los dos modos de funcionamiento, a los cuales el operario de mantenimiento puede acceder a través de la interfaz de usuario mostrada en la Figura 42. Estos modos son:

- Operación
- Mantenimiento

El modelo del SMC tiene capacidad para gestionar hasta 10 Sistemas Patrón.

Al iniciar el modelo del SMC, aparece una pantalla de configuración, ver Figura 41, a través de la cual podemos indicar los archivos y directorios que va a

utilizar el SMC para almacenar los datos de mantenimiento de los equipos en archivos XML, y los informes de mantenimiento generados en archivos HTML.

Además permite indicar fijar la fecha de instalación de los sistemas y el número de horas que la plataforma está en operación al día. Estos datos serán utilizados por la función de procesado estadístico de fallos.

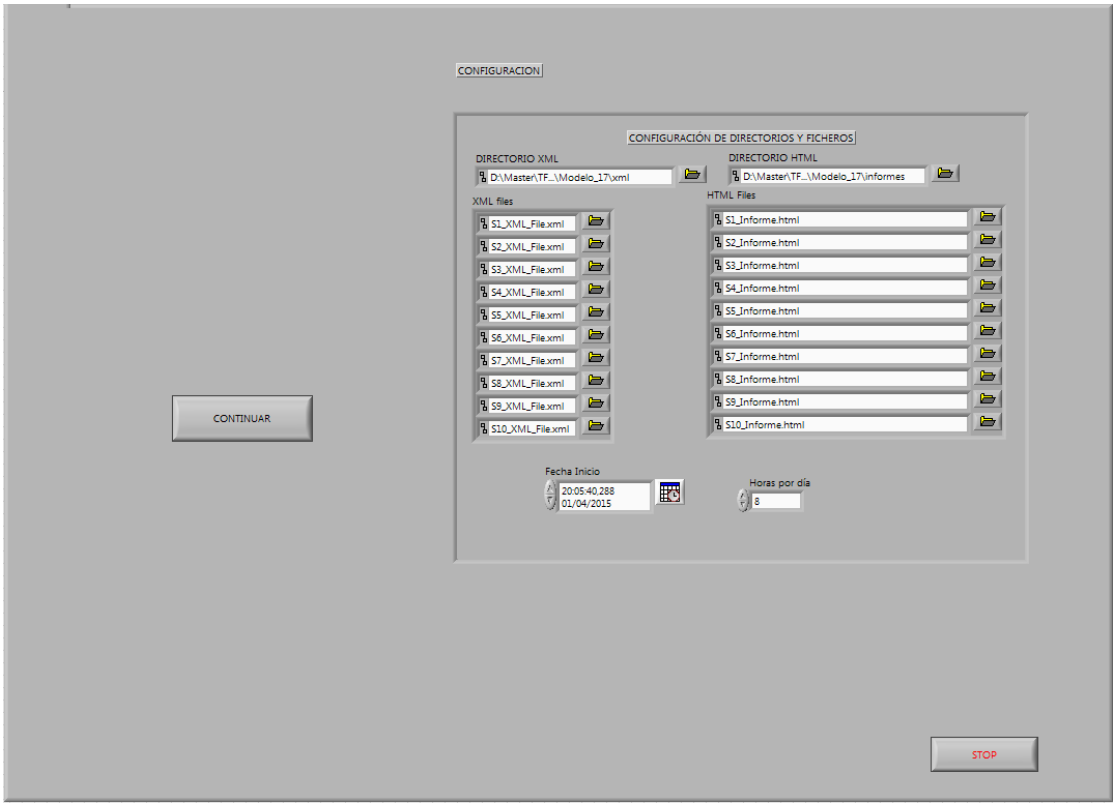


Figura 41. Interfaz SMC, página de Configuración.

Figura 42. Interfaz SMC, Menú principal.

6.4.2.1. Modo Operación

En el modo “Operación” el SMC proporciona al operario, información en tiempo real es estado de los sistemas el EMC. Esta información está accesible a través de la interfaz de usuario mostrada en el Figura 43.

La Figura 44, muestra el detalle de la información que el SMC muestra en el modo “Operación”. Para cada Sistema Patrón, el SMC indica de forma visual, si el Sistema Patrón está activo a través del indicador “Sistema On/Off”, además, indica que tipo de test está realizando el Sistema Patrón, si es un Test de Inicio (IBIT) o uno test continuo (CBIT) que tipo de test están realizando y si presentan algún fallo.

Cuando alguno de los Sistemas Patrón activos, reporta un fallo, la pantalla “Estado de los Sistemas” indica que el sistema está fallado, mientras el SMC registra la información del fallo recibida junto con la marca de tiempo en el archivo XML asociado al Sistema Patrón correspondiente.

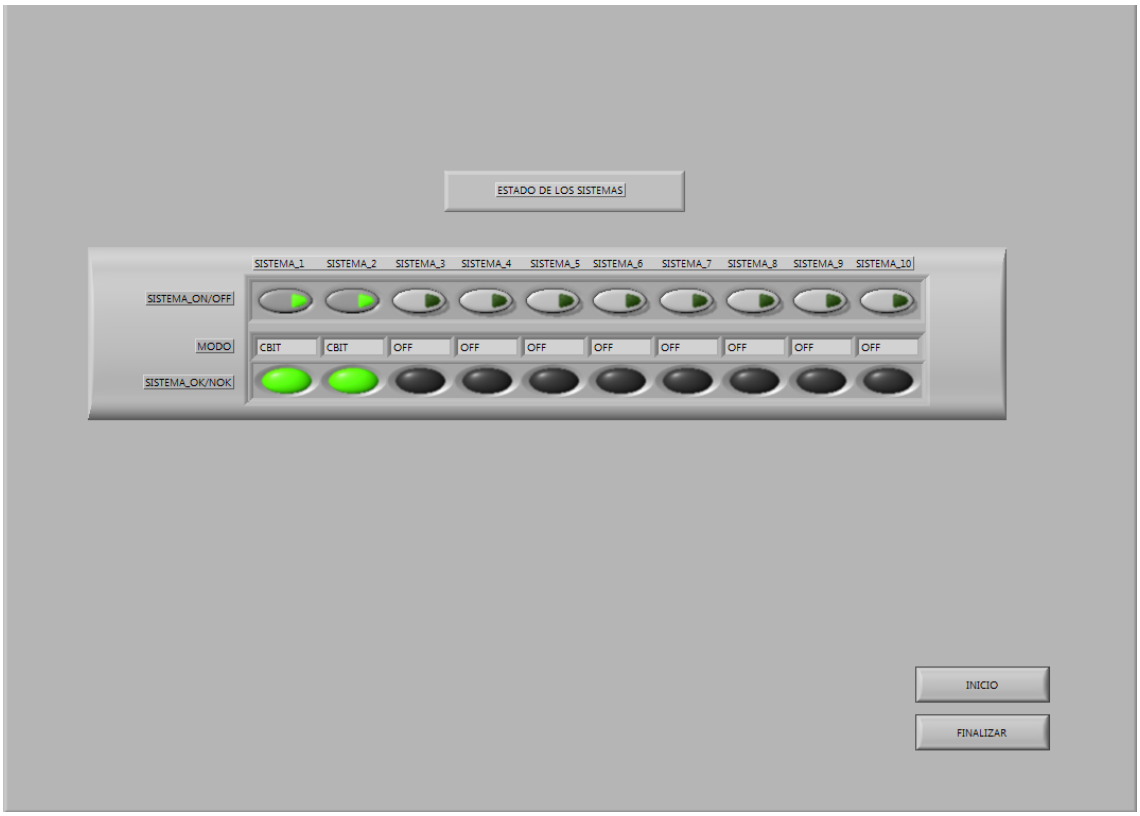


Figura 43. Interfaz SMC, página de Operación.

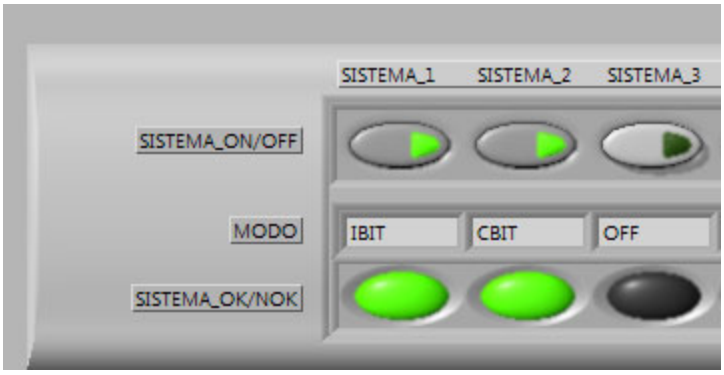


Figura 44. Interfaz SMC, detalle de la página de Operación.

6.4.2.2. Modo Mantenimiento

En el modo “Mantenimiento” el operario de mantenimiento, a través de la interfaz de usuario, puede acceder a las siguientes funciones, como indica en detalle la Figura 45:

Figura 45. Detalle de las opciones de Mantenimiento en la Interfaz SMC.

6.4.2.2.1. Fallos recibidos.

Al seleccionar la pestaña “Fallos Recibidos”, el SMC presenta la pantalla mostrada en la Figura 46, a través de la cual el SMC permite seleccionar el Sistema Patrón del cual se quiere mostrar los fallos que ha reportado. Adicionalmente, desde la ventana, se permite seleccionar entre el histórico de datos de fallo del Sistema Patrón seleccionado, o los datos de fallo para una fecha en concreto.

Tras la selección, el SMC lee el XML asociado al Sistema Patrón seleccionado y analiza los fallos que tiene registrados. De acuerdo con la selección realizada, el SMC presenta los datos de los fallos almacenados, en la pantalla ordenados por fecha, con la información mostrada en la Tabla X.

Fecha y hora
Sistema
Numero de parte del sistema
Número de serie del sistema
Tipo de test
Código de fallo
Módulo acusado 1
Módulo acusado 2
Módulo acusado 3

Tabla X. Información presentada para cada fallo.

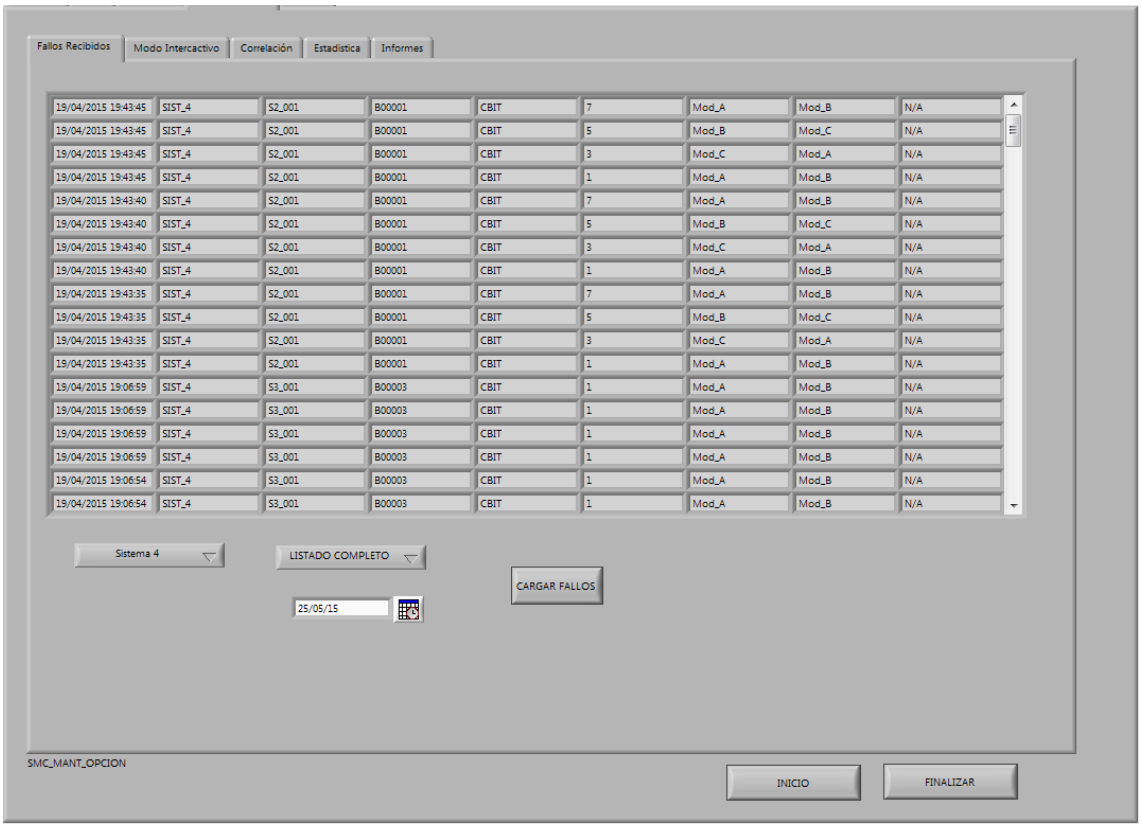


Figura 46. Interfaz SMC, Mantenimiento, página de Fallos Recibidos.

6.4.2.2.2. Modo Interactivo.

Al seleccionar, la pestaña de “Modo Interactivo”, el SMC presenta pantalla que muestra la Figura 47. A través de la cual el operador de mantenimiento selecciona el Sistema Patrón sobre el cual se quiere aplicar el Test Interactivo. De este modo el SMC envía el comando de Test Interactivo al Sistema Patrón seleccionado solicitando el Test Interactivo.

Cuando el SMC recibe el resultado del test, en caso de recibir algún fallo, marca en rojo el indicador de resultado del Test Interactivo y presenta en la ventana correspondiente la información del fallo o los fallos reportados. Ver Tabla X.

Con la información mostrada, el operador de mantenimiento debe realizar las tareas correspondientes sobre el Sistema Patrón fallado y volver a realizar el Test Interactivo para comprobar que no presente ningún fallo.

Cuando el Sistema Patrón está no reporta ningún fallo, el SMC marca en verde el indicador de estado.

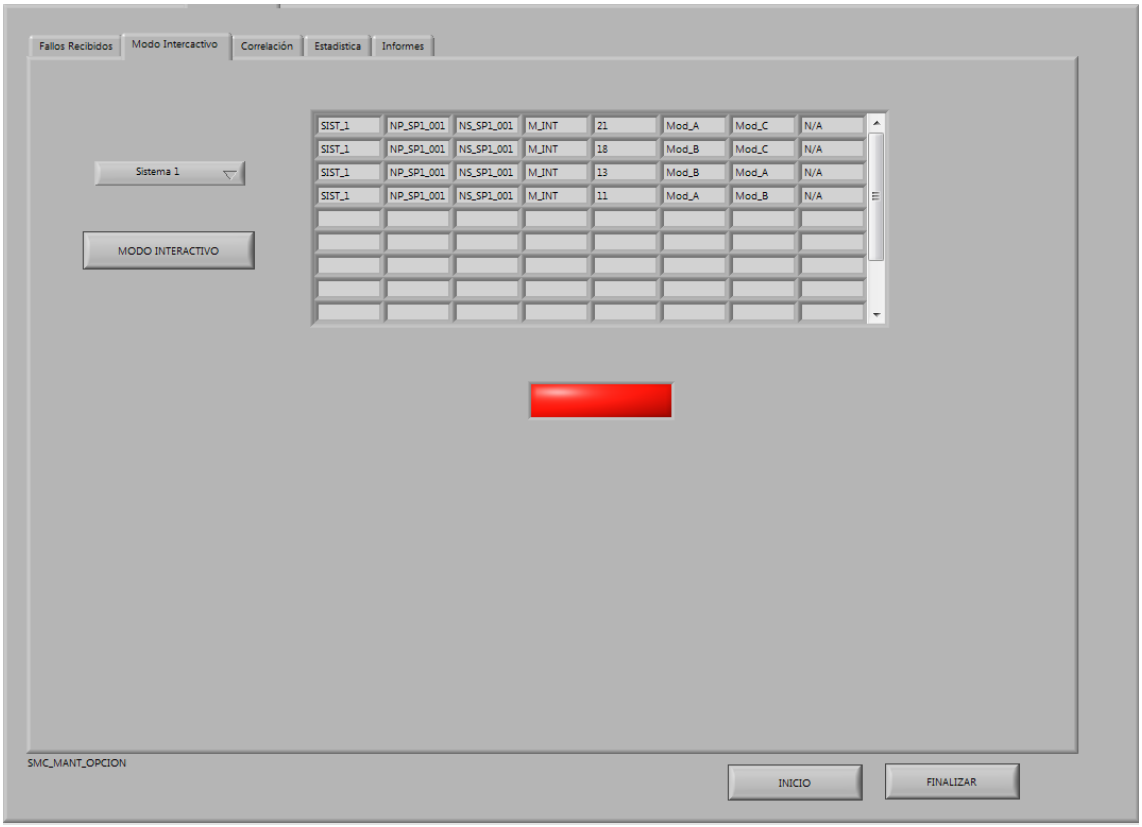


Figura 47. Interfaz SMC, Mantenimiento, página Modo Interactivo.

6.4.2.2.3. Correlación de fallos.

El modelo del SMC incluye una función de correlación o aislamiento de fallos para fallos de alimentación. A través de esta función el SMC comprueba si varios Sistemas Patrón, han reportado fallos de alimentación en un mismo minuto, lo cual puede indicar que el fallo de alimentación puede estar debido a un fallo de la alimentación global de la plataforma y no un fallo de cada sistema.

El SMC a través de la interfaz de usuario, que muestra la Figura 48, permite seleccionar la fecha en la cual se quiere realizar la correlación de fallos.

Cuando el operador de mantenimiento seleccionar “Calcular Correlación”, el SMC analiza la información almacenada de todos los sistemas para la fecha seleccionada.

El análisis de la información de fallos consiste en comprobar si hay fallos de alimentación y en qué instante de tiempo (hh/mm) se han producido. De forma gráfica el SMC presenta los sistemas que han reportado fallos de alimentación para una cierta hora.

Si hay más de un sistema que presente fallo de alimentación en el mismo instante de tiempo (minuto), se considera un fallo de la alimentación del sistema marcando en rojo el indicado “Alimentación Plataforma”.

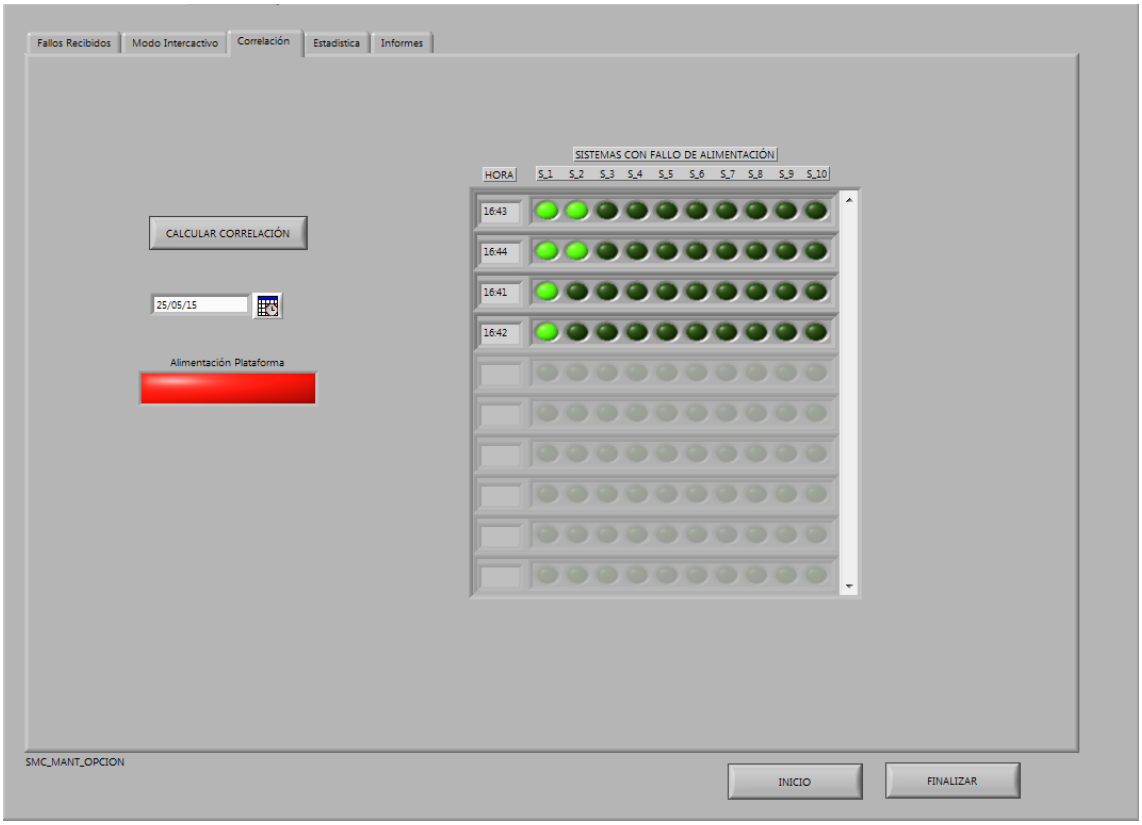


Figura 48. Interfaz SMC, Mantenimiento, página Correlación de fallos.

6.4.2.2.4. Procesado estadístico

El SMC dispone de una función de procesado estadístico, a través de la interfaz que muestra la Figura 49. Esta función estadística está basada en la aplicación de la distribución de Weibull sobre los datos almacenados por el SMC.

La aplicación de la distribución de Weibull se realiza sobre los fallos que tenga registrado un sistema determinado para uno de sus módulos, el cual ha podido ser acusado con tres niveles de prioridad.

A través de la interfaz del SMC la función estadística, el operador de mantenimiento realiza la selección sobre la que desea aplicar la distribución de Weibull: sistema, el módulo y prioridad. Además puede ajustar el nivel de fiabilidad mínimo que la plataforma requiere para el sistema seleccionado.

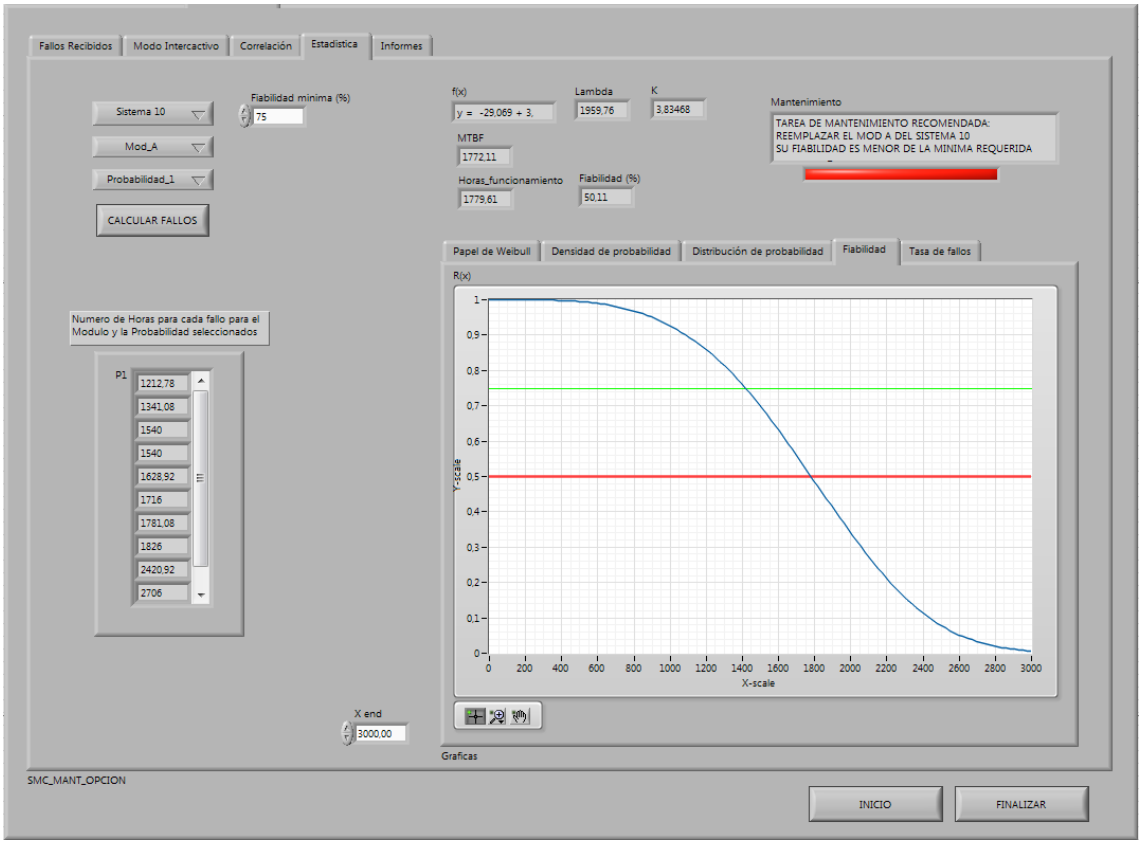


Figura 49. Interfaz SMC, Mantenimiento, página Estadística.

Tras la selección de estos parámetros, al pulsar el botón “CALCULAR FALLOS”, la función lee el fichero de fallos del archivo XML correspondiente al sistema seleccionado, e identifica los fallos que han acusado al módulo y prioridad seleccionados. Con esos datos la función calcula las horas que el sistema ha estado funcionando hasta cada fallo. Estos datos son presentados en una tabla, en la parte izquierda de la interfaz. Adicionalmente, la función también calcula el tiempo que lleva funcionando el equipo seleccionado desde el último fallo.

Con la información de las horas de funcionamiento hasta cada fallo para el modulo seleccionado, la función estadística calcula los parámetros de la

distribución de Weilbull, β y η, a través de la representación de los fallos y horas en el gráfico de Weibull, como explica el apartado 5.6.2.

En la interfaz de usuario en la parte superior se muestran la función de la recta de tendencia extraída del gráfico de Weibull junto con el valor de los parámetros β y η, los cuales, en el modelo de LabVIEW, se denominan K y Lambda respectivamente.

Con los parámetros característicos β y η obtenidos, la función estadística representa las gráficas de la distribución de Weibull:

- Densidad de probabilidad, f(x).
- Distribución de probabilidad, F(x).
- Fiabilidad, R(x).
- Tasa de fallos, λ(x)

La Figura 50 muestra, los menús para la selección de las diferentes gráficas calculadas con la distribución de Weibull.

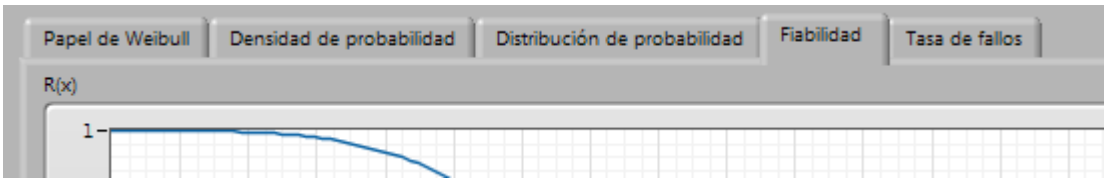


Figura 50. Menús de gráficas de la función Estadística.

Adicionalmente, la función estadística calcula el MTBF del equipo bajo análisis, junto con la fiabilidad del equipo instalado actualmente en función del número de horas que lleva operativo, desde el último fallo.

En la gráfica de Fiabilidad, R(x), también se representa una línea con el valor límite de la fiabilidad permitida para el equipo bajo análisis, junto con la fiabilidad del equipo instalado actualmente.

Si la fiabilidad del equipo actualmente instalado en la plataforma, es menor que la fiabilidad mínima requerida para dicho equipo, la interfaz presenta un aviso, indicando la recomendación de sustituir el módulo seleccionado.

6.4.2.2.5. Generación de informes.

El SMC proporciona una función para la generación de informes de mantenimiento en formato HTML [38]. A través de la interfaz que muestra la Figura 51, el operador de mantenimiento puede seleccionar el Sistema Patrón y la fecha sobre la cual se solicita el informe.

Al seleccionar la generación de informe, el SMC genera el informe correspondiente en la dirección especificada en la interfaz de configuración, Figura 41.

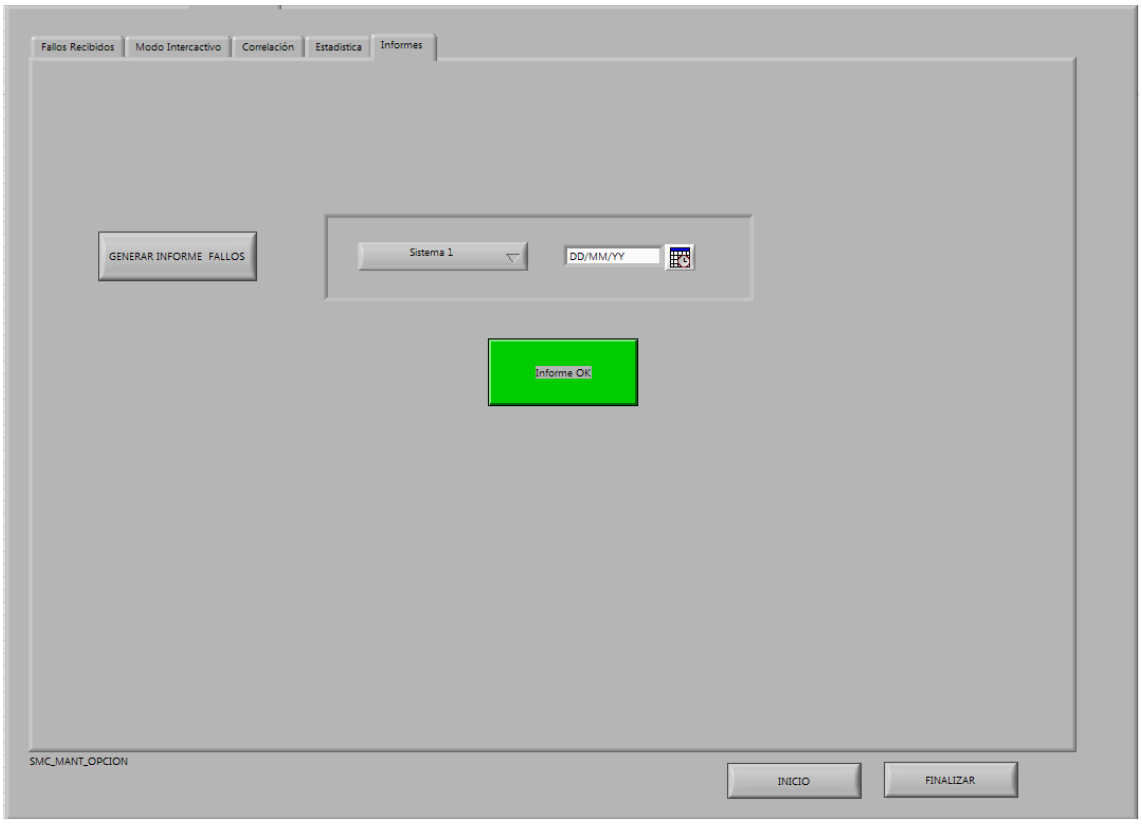
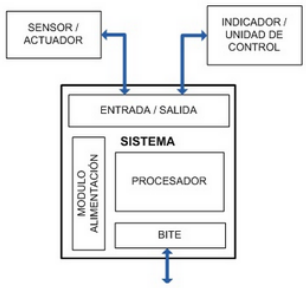


Figura 51. Interfaz SMC, Mantenimiento, página de Generación Informes.

En la Figura 52, muestra un ejemplo de informe de mantenimiento generado. El informe incluye el nombre del Sistema Patrón junto con un esquema de la arquitectura de dicho sistema. A continuación incluye los fallos del sistema para la fecha señalada y los datos correspondientes a cada fallo, (ver Tabla X), sobre los cuales son necesarias la realización de las tareas de

mantenimiento correspondientes. Por último el informe, incluye el histórico de fallos del sistema.

INFORME DE MANTENIMIENTO
SISTEMA 2



EL DÍA 25/05/2015 EL SISTEMA 2HA REPORTADO LOS SIGUIENTES FALLOS:

FECHA HORA	SISTEMA	NUMERO PARTE	NUMERO SERIE	MODO TEST	CODIGO FALLO	MODULO ACUSADO 1	MODULO ACUSADO 2	MODULO ACUSADO 3
25/05/2015 16:43:55	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:00	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:04	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:08	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:13	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:17	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:21	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 16:44:26	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A

ACCIÓN DE MANTENIMIENTO: REALIZAR LAS TAREAS DE MANTENIMIENTO ASOCIADAS A LOS FALLOS REPORTADOS ANTERIORMENTE.

HISTORICO DE FALLOS REPORTADOS POR EL SISTEMA

FECHA HORA	SISTEMA	NUMERO PARTE	NUMERO SERIE	MODO TEST	CODIGO FALLO	MODULO ACUSADO 1	MODULO ACUSADO 2	MODULO ACUSADO 3
25/05/2015 17:21:54	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	18	Mod_B	Mod_C	N/A
25/05/2015 17:21:49	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	18	Mod_B	Mod_C	N/A
25/05/2015 17:21:49	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	10	Mod_A	Mod_B	N/A
25/05/2015 17:21:49	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 17:21:45	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	18	Mod_B	Mod_C	N/A
25/05/2015 17:21:45	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	10	Mod_A	Mod_B	N/A
25/05/2015 17:21:45	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	5	Mod_B	Mod_C	N/A
25/05/2015 17:21:45	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	2	Mod_B	Mod_C	N/A
25/05/2015 17:21:41	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	18	Mod_B	Mod_C	N/A
25/05/2015 17:21:41	SIST_2	NP_SP2_001	NS_SP2_001	CBIT	10	Mod_A	Mod_B	N/A

Figura 52. Informe generado.

6.4.3. Limitación del protocolo de comunicaciones TCP/IP en el modelo.

El modelo de EMC desarrollado en LabVIEW consta de diferentes Sistemas Patrón (clientes) y un SMC (servidor). Al ejecutar todas las partes del modelo en una misma máquina (PC), no es posible identificar cada sistema modelado por su IP o “Host Name”.

Esta particularidad, afecta al SMC, en la función de Modo Interactivo impidiendo identificar a que Sistema Patrón se le comanda un Test Interactivo desde el SMC.

Debido a esta limitación, aunque el modelo de EMC tiene implementado una comunicación bidireccional a través de TCP/IP, con la capacidad de enviar comandos de Modo Interactivo al Sistema Patrón correspondiente, el modelo funciona del siguiente modo:

Cuando se selecciona la función Modo Interactivo desde el SMC, para un Sistema Patrón, dicha función utiliza la información recibida del Sistema Patrón en Modo Continuo, presentándola como respuesta al Modo Interactivo.

7 Evaluación, conclusiones y trabajos futuros

7.2. Evaluación

En este capítulo se evalúa el cumplimiento de los objetivos presentados al inicio de este TFM. La Tabla XI muestra los principales objetivos iníciales del TFM y su cumplimiento.

	Objetivos iníciales	**Cumplimiento**
1	Presentar aproximación para el mantenimiento centralizado de sistemas complejos	✓
2	Implementar en LabVIEW una aproximación de un Entorno de Mantenimiento Centralizado de sistemas complejos basada en BITE	✓
3	Integrar en el Entorno de Mantenimiento Centralizado de LabVIEW una función estadística para el diagnóstico y aislamiento de fallos.	✓
4	Proporcionar una visión integral de las arquitecturas de sistemas	✓
5	Analizar los métodos y técnicas de mantenimiento	✓
6	Revisar las principales técnicas de BITE	✓
7	Presentar los beneficios de un entorno de mantenimiento centralizado	✓

Tabla XI. Objetivos iníciales del TFM.

En relación al objetivo número 2, la implementación de un Entorno de Mantenimiento Centralizado, se han comparado los resultados arrojados por el modelo realizado en LabVIEW con los resultados aportados con su sistema de diagnóstico centralizado de un avión de transporte moderno. En concreto, el modelo LabVIEW responde a una modelización a nivel de caja negra del sistema de diagnóstico mencionado anteriormente. También se ha utilizado el sistema de diagnóstico centralizado de un avión de transporte moderno para la evaluación del modelo. Más en detalle, y para los siguientes aspectos:

- Detección de fallos:

  El modelo incluye una representación para algunos de los fallos que pueden mostrar un sistema de avión. Para los fallos modelizados en el Sistema Patrón, el modelo del SMC es capaz

de detectar los fallos introducidos en el Sistema patrón y enviar su correspondiente código de fallo a través del mensaje correspondiente, de forma similar al funcionamiento de un sistema de avión.

- Aislamiento de fallos:

  Para el aislamiento de fallos, el modelo incluye la función Modo Interactivo. Esta función opera de acuerdo la limitación descrita en el apartado 6.4.3, debida a la ejecución de los diferentes sistemas patrón y el SMC desde un único procesador.

- Correlación de fallos:

  El modelo de LabVIEW, incluye una modelización de la correlación de fallos de alimentación de los sistemas. El comportamiento de esta función, es similar al que presenta las correlaciones de fallos entre sistemas de avión.

- Generación de Informe:

  El modelo incluye una representación de algunas de las capacidades de generación de informes de mantenimiento de las que consta un sistema de avión y que previsiblemente, se extenderán a cualquier plataforma industrial que implemente un EMC. Para la elaboración de este informe se ha tenido en cuenta que debe ser fácilmente inteligible por el personal encargado del mantenimiento

En relación al objetivo número 3, a pesar de que el procesado estadístico de datos de fallos es una práctica extendida en la ingeniería de mantenimiento, hasta ahora, no se ha aplicado en los sistemas de mantenimiento centralizados en la industria. Como demuestra la integración de esta función en el modelo objeto de este TFM, el análisis estadístico de los datos de fallo de un equipo, presenta una ayuda importante al mantenimiento predictivo.

7.3. Trabajos Futuros

El desarrollo de este TFM, da opción a la realización de futuros trabajos en línea con los conceptos introducidos e implementados hasta ahora, como pueden ser:

- Ampliación de funcionalidades del CMS:

  Incluyendo las funcionalidades más comunes de la industria aeronáutica, ferroviaria y del automóvil, como puede ser la gestión de configuración de la versión de cada uno de los sistemas.

- Ampliación de capacidades BITE del Sistema Patrón:

  Incluyendo la funcionalidad automática de BITE del Sistema Patrón, e incluir la detección de fallos en las interfaces entre los distintos módulos del Sistema Patrón En este proyecto se ha considerado únicamente la inyección manual de fallos.

- Aplicaciones con Meta-Sistemas:

  Ampliando el modelo e incluyendo la interfaz con otros modelos peer-to-peer y validar la interfaz con otros sistemas.

- Filtro de fallos espurios:

  Incluyendo la capacidad de filtrar fallos espurios, a través de la integración de un filtro, cuya configuración pueda ser realizada por el usuario, a través de un UMS [3].

- Tests de no-regresión:

  Incluyendo en el modelo la ejecución automática de test que permitan asegurar que una nueva versión ofrece, al menos, la misma funcionalidad que la versión anterior, y libre de errores.

7.4. Conclusiones

A lo largo de este TFM se ha introducido y desarrollado el concepto de mantenimiento aplicado no solo a un equipo o a un sistema, sino a una plataforma multisistema y metasistema, de forma integral, aportando el diseño e implementación de una solución para este concepto, lo que ha permitido, tras la realización de este TFM, establecer las siguientes conclusiones obtenidas:

El mantenimiento debe ser uno de los requisitos funcionales principales a la hora de plantear el diseño de un sistema mantenible. Por lo tanto debe de tenerse en cuenta desde el comienzo del diseño y extenderse todo el ciclo de vida del sistema.

Partiendo del hecho de que hoy en día los sistemas desarrollados y utilizados por las diferentes industrias son cada vez son más complejos, con un mayor nivel de integración, y que requieren de un mantenimiento específico, complejo, y por lo tanto automatizado. La implementación de funciones BITE en los sistemas se ha convertido en una herramienta básica para el mantenimiento de las distintas plataformas industriales.

Teniendo en cuenta que en las diferentes industrias cada vez es más común la utilización de plataformas formadas por sistemas complejos, la integración de un sistema de mantenimiento estandarizado y centralizado para estas plataformas, se está convirtiendo en un requisito fundamental en el diseño y desarrollo de las mismas.

La clave para implementar el Sistema de Mantenimiento Centralizado desarrollado en este TFM en una plataforma industrial multisistema reside en la estandarización de las capacidades BITE para todos los sistemas incluidos en dicha plataforma, al igual que ocurre en industrias en las que estos sistemas están más desarrollados (aeronáutica, ferroviaria, etc.).

Pese a que inicialmente el planteamiento de un entorno centralizado de mantenimiento en una plataforma industrial multisistema pueda representar un porcentaje elevado del coste total de la plataforma, los beneficios resultantes de la implementación de un entorno de mantenimiento centralizado, lo convierten en una inversión rentable para la gran mayoría de las diferentes industrias, por

extrapolación de su aplicación en las áreas aeronáutica, del automóvil y ferroviaria principalmente.

Por último, la formalización y estandarización del diseño propuesto, así como el tratamiento estadístico incluido en este TFM, permiten que la solución aportada pueda aplicarse directamente a diferentes industrias que utilizan plataformas multisistemas, incluyendo la aeronáutica, ferroviaria y automovilística, y por extensión, a cualquier plataforma industrial integrada por diferentes sistemas.

7.4.1. Aportaciones

La principal aportación de este proyecto es la modelización y estandarización de las funciones de detección y aislamiento automática de fallos, que permiten la identificación rápida, fiable y segura de los fallos de funcionamiento de los sistemas de una plataforma, permitiendo también la generación automatizada de informes de mantenimiento, lo cual lleva consigo una disminución de los costes de mantenimiento y el aumento del tiempo de operación de los sistemas de una plataforma, al disminuir los tiempos dedicados al mantenimiento.

Igualmente, la modelización del entorno de mantenimiento centralizado, permite conocer los resultados del sistema sin necesidad de construirlo, lo que permite acortar tiempos de desarrollo y abaratar costes de fabricación, al tiempo que incrementa la calidad (al descubrir fallos de diseño antes de fabricar el sistema) y la funcionalidad de los sistemas a construir.

Además, el tratamiento estadístico de los fallos permite un sistema predictivo de las actividades de mantenimiento, de modo que se aumenta la fiabilidad del sistema, al reemplazar el componente del sistema antes de que se produzca su fallo, en contraposición al mantenimiento correctivo, lo que permite acomodar de forma programada los tiempos de mantenimiento a los períodos de no operación del sistema.

Adicionalmente, la estandarización como sistema modular y su amplia aplicación a cualquier plataforma industrial permitiría abaratar considerablemente los costes de implementación, fabricación, entrenamiento y uso.

8 Glosario

- Analizador de resultados de test:

  El analizador de los resultados de test, normalmente es un comparador que analiza las respuestas de los test y las compara con los resultados de referencia generados o almacenados en el analizador. El analizador de los resultados de test debe indicar si el circuito bajo test funciona correctamente o si por el contrario, tiene algún elemento fallado.

- Aislamiento de fallos:

  Es el procedimiento necesario para aislar un fallo concreto. Estos procedimientos son los que se implementan en los diferentes test del BITE.

- BIT continuo (CBIT):

  Son los test que de forma continua están monitorizando el funcionamiento del sistema. En caso de que detecte algún fallo, este es reportado al sistema central de mantenimiento.

- BIT Inicial (IBIT):

  Test de comprobación del sistema que se ejecuta automáticamente cada vez que se energiza el sistema. Este test tiene la capacidad de indicar si el sistema está operativo o no.

- BITE (Built-In-Test Equipment):

  Es la capacidad de un equipo de realizar de forma automática test de autocomprobación para detectar si el propio equipo presenta algún fallo.

- Cobertura del BITE:

  La cobertura del BITE en un sistema es la relación entre la parte que puede ser comprobada por un grupo de test y el total del sistema. Un test en un sistema puede comprobar que una función de sistema funciona, pero no garantiza que funcionen todas las funciones del sistema.

Para definir la cobertura en un sistema es necesario definir una estrategia de test, mediante la cual se fija que parte del sistema es necesario probar.

- Controlabilidad:

  Es la medida de cómo de difícil es excitar con un valor en un punto de un circuito para excitar un fallo determinado [1], [31].

- Despachabilidad:

  Es la capacidad de poner una plataforma en operación, dependiendo de los fallos que reporta y de acuerdo a la regulación aplicable. Este concepto se aplica principalmente en aviación.

- Detección de fallos:

  Proceso por el cual se determina si un equipo o sistema tiene algún fallo. La detección de fallos se basa en estrategias BITE.

- Diagnosis:

  Es la capacidad de identificar un fallo en un sistema. Fallo en un sistema o equipo:

  Estado de incapacidad del equipo o sistema para realizar su función.

- Efectividad del BITE:

  Es la medida de la capacidad del BITE de un equipo para detectar los fallos que puede tener.

- Fallo No Encontrado (NFF):

  Circunstancia ante la cual un sistema reporta un fallo a través de su BITE pero en la reparación no se encuentra ningún fallo.

- Generador de patrones de test:

  Es el dispositivo que tiene almacenados los diferentes patrones de test para la realización de los test en un circuito. Normalmente estos dispositivos son memorias ROM [31].

- Mantenibilidad:

  Es la capacidad de un equipo de que se mantenga operativo siempre que se respeten los procedimientos de mantenimiento definidos para él.

- Mantenimiento de línea:

  En un entorno de mantenimiento centralizado, el mantenimiento en línea tiene como funciones principales en la identificación de un fallo, el aislamiento de ese fallo a nivel de equipo, el reemplazo del equipo fallado, los ajustes y configuración, y comprobaciones necesarias para poner el sistema en operación.

- Observabilidad:

  Es la medida de cómo es de difícil propagar una señal fallada desde una línea del circuito a un punto de salida del circuito [1], [31].

- Patrón de test:

  Son las condiciones necesarias predefinidas para estimular un circuito, con el fin de comprobar un caso de test particular [31].

- Relación entre Cobertura y Diagnosis:

  La cobertura y la diagnosis están estrechamente relacionadas. No se puede alcanzar un diagnóstico total en un sistema, si la cobertura del conjunto de test no es total [1].

  Por lo tanto, si en un sistema, el conjunto de test del BITE no garantiza la suficiente cobertura, es muy probable que el diagnóstico proporcionado por este BITE no sea del todo real.

Por el contrario, si en un equipo se requiere un nivel de diagnóstico a nivel tarjea o módulo interno, no es necesario que el BITE sea capaz de identificar el fallo a nivel componente. En este caso la excesiva capacidad de diagnóstico no está justificada.

Por lo tanto para definir la relación entre la Cobertura y la Diagnosis, es requisito imprescindible diseñar una estrategia de test y establecer el nivel de diagnóstico.

- Sistemas tolerantes a fallos:

  Por diseño, los sistemas deben tener la capacidad de continuar con su operación en caso de presentar alguno de los fallos para los que son tolerantes.

9 Bibliografía

[1] ARINC REPORT 612 Built-In Test Equipment glossary.

[2] F. J. Gonzalez, Teoría y práctica del mantenimiento industrial avanzado, Fundación Confemetal, 2005.

[3] M. Sánchez-Puebla, R. Sobrino, J. Martín, “Everyday tools used for Avionics User Modifiable Software automatic generation”, 2015, IS-EUD 2015, 5th Symposium on End-User Development: My world, my device, my program.

[4] Maintenance Steering Group 3 - Task Force (MSG3), Maintenance Program Development Document, Rev. 2. Air Transport Association of America, 1993.

[5] J. Smith, D. Lowenstein, “Built in Test - Coverage and Diagnostics, Best Practices to Achieve Built in Test Success ”. IEEE. 2009.

[6] J. Smith, D. Lowenstein, “Built in Test - Coverage and Diagnostics, You can"t do one without the other if you want to be successful”. IEEE. 2009.

[7] ARINC REPORT 604-1 Guidance for design and use of Built-In Test Equipment.

[8] ARINC REPORT 624-1 Design Guidance For Onboard Maintenance System.

[9] ARINC SPECIFICATION 429 - MARK 33 Digital Information Transfer System (DITS).ARINC 724B Aircraft Communications Addressing and Reporting System (ACARS).

[10] ARINC REPORT 629 Multi-Transmitter Data Bus.

[11] ARINC REPORT 636 Onboard Local Area Network (OLAN).

[12] ARINC REPORT 646 Ethernet Local Area Network (ELAN)

[13] IEC 62580-1: Railway applications - Onboard multimedia and telematic subsystems for railways, 2013

[14] SAE J1962 - Diagnostic Connector Equivalent to ISO/DIS 15031-3.

[15] SAE J1850 - Class B Data Communications Network Interface.

[16] ISO 13400 Road vehicles -- Diagnostic communication over Internet Protocol (DoIP).

[17] ISO 14230 Road vehicles -- Diagnostic systems -- Keyword Protocol 2000.

[18] ISO 22901 Road vehicles -- Open diagnostic data exchange (ODX).

[19] ISO 27145 - Road vehicles - Implementation of World-Wide Harmonized On-Board Diagnostics (WWH-OBD) communication requirements.

[20] ISO 14229 Road vehicles -- Unified diagnostic services (UDS).

[21] ISO/IEC 7498-1 Tecnología de la información. Interconexión de sistemas abiertos. Modelo de referencia básico: el modelo básico.

[22] A. S. Tanenbaum, Computer_Networks, Prentice Hall, 2003.

[23] C.B. Watkins, “Transitioning from federated avionics architectures to Integrated Modular Avionics”, Digital Avionics Systems Conference, 2007. DASC '07. IEEE/AIAA 26th.

[24] DO-297 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations.

[25] ARINC Report 653 Avionics Application Software Standard Interface.

[26] RTCA/DO-178C Software Considerations in Airborne Systems and Equipment Certification, 2012.

[27] EN-13306 - Terminología del mantenimiento, 2001.

[28] AC-121-22A, Advisory Circular: Maintenance Review Boards, Maintenance Type Boards, and OEM/TCH Recommended Maintenance Procedures, FAA, 2012.

[29] Built in Test – Design and Optimization Guidelines. Office of the Assistant Secretary of the Navy. 2001.

[30] P. E. Lanigan, S. Kavulya, P. Narasimhan, “Diagnosis in Automotive Systems: A survey”. 2011.

[31] J. Rajski, J. Tyszer, “Arithmetic Built-in Self-Test for Embedded Systems,” Prentice Hall, PTR, October 1997.

[32] A. Birolini, Reliability Engineering: Theroy and Practice, Springer, 2014.

[33] RFC 793: Transmission Control Protocol.

[34] Basic TCP/IP Communication in LabVIEW, National Instruments, 2006.

[35] Internet Developers Toolkit for G: Reference Manual, National Instruments, 1998.

[36] National Instruments, “A Multi-client Server Design Pattern Using Simple TCP/IP Messaging”, 2011.

[37] A. Skonnard, “Essential XML Quick Reference”, Addison-Wesley, 2002.

[38] HTML5, A vocabulary and associated APIs for HTML and XHTML, 2014, http://www.w3.org/TR/html5.